

**Numerical Simulation of Ballistic Limit Curves  
for Orbital Debris Shielding**

**by**

**Robert James Rabb, B.S.**

**Thesis**

Presented to the Faculty of the Graduate School of  
The University of Texas at Austin  
in Partial Fulfillment  
of the Requirements  
for the Degree of

**Master of Science in Engineering**

**DISTRIBUTION STATEMENT A**

**Approved for public release;  
Distribution Unlimited**

**The University of Texas at Austin  
May 1998**

**19980514 036**

**DTIC QUALITY INSPECTED 3**

Copyright  
by  
Robert James Rabb  
1998

DTIC QUALITY INSPECTED 8

**Numerical Simulation of Ballistic Limit Curves  
for Orbital Debris Shielding**

Approved by  
Supervising Committee:

Eric P. Fahrnerthly

Alfred E. Fowler

## **Acknowledgements**

This work was supported in part by the Space Science Branch of NASA Johnson Space Center. Computer time was provided by the Texas Advanced Computing Center at the University of Texas at Austin.

April 28, 1998

## **Abstract**

# **Numerical Simulation of Ballistic Limit Curves of Orbital Debris Shielding**

by

Robert James Rabb, M.S.E.

The University of Texas at Austin, 1998

Supervisor: Eric P. Fahrenthold

EXOS is a recently developed particle-finite element code well suited to modeling hypervelocity impact phenomena. A series of EXOS simulations were conducted to evaluate the use of this new code for orbital debris shielding design. Two sets of simulations, one for a single bumper Whipple shield and one for a two bumper or stuffed Whipple shield, were compared to well known equations for the ballistic limit of Whipple shields, derived from experiment. The results show that EXOS provides an accurate and computationally tractable approach to simulate orbital debris shield performance.

## Table of Contents

List of Tables .....	viii
List of Figures .....	ix
Chapter 1: Introduction .....	1
Chapter 2: Literature Review .....	5
2.1 Hydrocodes .....	6
2.1.1 Eulerian Methods .....	8
2.1.2 Lagrangian Methods .....	9
2.1.3 ALE Methods .....	10
2.1.4 Particle Methods .....	12
2.1.5 PIC Methods .....	14
2.1.6 Mixed Methods .....	14
2.2 Particle Codes .....	16
2.3 Application of Particle Codes .....	18
2.3.1 Linking Codes .....	20
2.3.2 Multi-Bumper Designs .....	22
2.4 Summary .....	24
Chapter 3: Whipple Shields .....	26
3.1 Whipple Shield Ballistic Limit Equations .....	26
3.2 Methodology .....	28
3.3 Comparison of EXOS Simulations to the Ballistic Limit Curve .....	29
Chapter 4: Stuffed Whipple Shields .....	33
4.1 Stuffed Whipple Shield Ballistic Limit Equations .....	33
4.2 Methodology .....	36
4.3 Comparison of EXOS Simulations to the Ballistic Limit Curve .....	37

Chapter 5: Conclusions .....	40
Appendix A: EXOS Input File.....	69
Appendix B: Rezoner Input File .....	70
Bibliography.....	71
Vita .....	74

## List of Tables

Table 1: Projectile Diameters and Velocities Used for the 15° Impact Whipple Shield Simulations.....	42
Table 2: Projectile Diameters and Velocities Used for the 45° Impact Whipple Shield Simulations.....	43
Table 3: Material Properties Used in the Whipple Shield Impact Simulations ....	44
Table 4: Damage Predicted by the 15° Whipple Shield Simulations.....	45
Table 5: Damage Predicted by the 45° Whipple Shield Simulations.....	46
Table 6: Impact Simulation Time on the Cray J90 for the 15° Simulations .....	47
Table 7: Impact Simulation Time on the Cray J90 for the 45° Simulations .....	48
Table 8: Projectile Diameters and Velocities Used for the 1° Impact Stuffed Whipple Shield Simulations.....	49
Table 9: Material Properties Used in the Stuffed Whipple Shield Impact Simulations .....	50
Table 10: Damage Predicted by the Stuffed Whipple Simulations.....	51
Table 11: Impact Simulation Time on the Cray J90 for the Stuffed Whipple Simulations .....	52



## List of Figures

Figure 1: Whipple Shield Geometry .....	53
Figure 2: Ballistic Limit Curve for the 15° Whipple Shield Simulations .....	54
Figure 3: Ballistic Limit Curve for the 45° Whipple Shield Simulations .....	55
Figure 4: 15° EXOS Result for Simulation BL7.....	56
Figure 5: Side View of 15° EXOS Result for Simulation BL7.....	57
Figure 6: 15° EXOS Result for Simulation BL10.....	58
Figure 7: Side View of 15° EXOS Result for Simulation BL10.....	59
Figure 8: 45° EXOS Result for Simulation BLO9.....	60
Figure 9: Side View of 45° EXOS Result for Simulation BLO9.....	61
Figure 10: 45° EXOS Result for Simulation BLO6c .....	62
Figure 11: Side View of 45° EXOS Result for Simulation BLO6c .....	63
Figure 12: Stuffed Whipple Shield Geometry.....	64
Figure 13: All Aluminum Stuffed Whipple Shield Geometry .....	65
Figure 14: Ballistic Limit Curve for the 15° Stuffed Whipple Shield Simulation...	66
Figure 15: Stuffed Whipple Shield EXOS Result for Simulation SW8.....	67
Figure 16: Side View of Stuffed Whipple Shield EXOS Result for Simulation SW8 .....	68

## Chapter 1: Introduction

Orbiting space structures face the constant and growing threat of impact with orbiting space debris and meteoroids. After every mission, inspections of Space Shuttle Orbiters disclose some impact damage to the unprotected surfaces. The frequencies of these impacts suggest that the population of small orbital debris is increasing, due to the impacts of the debris with spacecraft as well as other orbital debris. For spacecraft such as satellites and the upcoming International Space Station, which are subject to long duration space missions in low Earth orbit, the probability of a catastrophic impact increases.

An effective debris shield must protect the spacecraft from orbital debris and micrometeoroid impacts. The greatest danger from orbital debris is from fragments one millimeter to one centimeter in length with an average velocity of 13 km/s and density of  $2.8 \text{ gm/cm}^3$ . There are larger fragments but the probability of a collision with debris over one centimeter is remote (Isbell et al., 1993). The micrometeoroid threat is a result from dust-size particles with an average velocity of 20 km/s. The man-made space debris threat is more of a hazard than the micrometeoroid menace because of its larger mass and particle size (Hertel, 1993). These orbital debris properties push the region of interest beyond the velocity range of conventional experimental techniques.

Research at the NASA Johnson Space Center (JSC) Hypervelocity Impact Technology Facility has resulted in several new shielding geometries and materials that provide improved protection and weight savings over conventional all-

aluminum Whipple shields. Most previous orbital shielding design research has been experimental, with analytical methods used to predict impact damage outside of experimental capabilities. However, computer simulations with particle codes have evolved in providing realistic estimates of impact damage. There are several reasons for the use of particle codes in orbital debris shielding design:

- (1) new materials and shield configurations have increased the quantity of competing shield designs,
- (2) researchers must conduct a large number of experiments to distinguish the three dimensional performance of an independent design concept due to the complex interaction of material failure effects, impact obliquity, and impact velocity,
- (3) quicker, less expensive spacecraft shielding design places importance on simulation and not experiment,
- (4) the region of impact velocities and kinetic energies of interest is beyond the capabilities of conventional light gas guns, and
- (5) some physical parameters (temperature, pressure, etc.) cannot be accurately obtained in such short duration experiments.

The factors listed above show that improved computer codes will assume a larger role in orbital debris shield design, even though experimental research will still have a crucial purpose. Supercomputers and streamlined codes have their limitations, and simulation results must be validated with experiment when possible. Current Eulerian and Lagrangian hydrocodes are not appropriate for shielding design simulation, requiring augmentation for three dimensional debris

cloud propagation. Pure particle codes address the three dimensional debris cloud dynamics of shielding design, but particle codes have shortcomings in treating some strength and boundary effects. Particle codes may not accurately predict the structural response to impact (Fahrenthold, 1995).

Acknowledging the benefits of particle methods, several coupled particle-finite element codes for orbital shielding design application have emerged. One of these codes is EXOS (Fahrenthold, 1997), a hybrid particle-finite element code, with a rezoning option. Rezoning allows the user to specify the region of interest within a shielding configuration and ignores the particles that have migrated out of this region, optimizing computation time. Appendix A shows an example EXOS input file, and Appendix B shows an example rezoner file.

Another tool researchers use for orbital shielding design is the ballistic limit curve. The ballistic limit curve is obtained from experimental data and graphically portrays the projectile diameter and velocity threshold at which a specific shielding configuration defeats a given projectile. Failure of a shield is when the shield's rear wall is visibly breached or spalled. The ballistic limit for a shield configuration is a function of many parameters including projectile diameter, velocity, impact obliquity, density, and shield density. Ballistic limit curves graphically describe a shield's performance and are used to assess the overall risk for spacecraft.

The purpose of this thesis is to evaluate the application of a particle-finite element code (EXOS) for orbital debris shielding design. An array of three dimensional simulations involving oblique impact on single and multi-plate aluminum shielding designs has been conducted for comparison to published

experimental data. The first section of this paper discusses previous research in the use of computer models for hypervelocity impact simulation. Results of Whipple shield impact simulations at impact obliquities of  $15^\circ$  and  $45^\circ$ , at various projectile diameters and impact velocities, are reported and compared to analytic ballistic limit curves derived from experiment. Next, results of stuffed Whipple shield simulations are documented for various projectile sizes and impact velocities and compared to experiments. Finally, the results of all simulations are compared to published ballistic limit equations, and memory and CPU requirements for the simulation of example problems are given.

## **Chapter 2: Literature Review**

Early research in hypervelocity impacts employed electromagnetic guns which were incapable of producing true hypervelocities. However, by 1957, the electromagnetic gun lost ground to the light gas gun which could achieve velocities up to 10 km/s. Light gas gun experimentation led to the development of equations of state to describe the behavior of materials under high pressure and temperature conditions found in hypervelocity impacts. Although the size of light gas guns has prohibited their use outside the laboratory, the light gas gun has served as the most dependable launch device for experimental hypervelocity impact research.

Hypervelocity research waned during the 1960's as speculations about meteoroids and moon dust were exaggerated. Then during the 1970's and 1980's, hypervelocity research resurfaced with new applications. There were new interests in space colonization, electromagnetic launch, strategic defense, armor / antiarmor, planetary science, and fusion research.

Although the military motivation for armor / antiarmor applications had the most near term potential for hypervelocity studies, the interest in impact damage of spacecraft also reappeared. Over 30 years of launches and tests had generated a large amount of space debris. Recently, encounters with hypervelocity space junk have scarred the U.S. shuttle. Debris and shrapnel from launches and tests have created a particle belt which grows as collisions among the pieces of space junk generate even more debris. Fortunately, the supporting technologies have now

matured to where higher velocities of macroscopic particles are achievable in the laboratory, and computational devices can assist and shorten the time to model the impacts (Fair, 1987).

## **2.1 HYDROCODES**

Hydrocodes are large computer programs that can numerically emulate highly dynamic events by solving the conservation equations and accounting for material variations. These programs simulate the propagation of the shock waves and compute the stresses, strains, velocities, and other parameters as a function of time and position. Hydrocodes can predict the response of a penetrator and target, providing insight and explaining phenomena. They can extend experimental data by performing variation of parameters, be used as an alternative to experiments, and be the only resource in extremely high velocity experiments.

Although computers make numerical modeling possible, they have limitations; memory requirements and calculation time must be reasonable. The efficiency of numerical schemes can be interpreted as the total number of arithmetic, logical, and storage operations performed by the computer over a unit time to obtain a solution. Efficiency decreases with greater complexity of the numerical technique, but the accuracy of the solution generally increases with increasing complexity. Hydrocodes compute several thousands of variables for each time step. It is imperative that the numerical differencing and integration technique be as efficient as possible, but it must be accurate as well for the solution to be of any significance. There must be a compromise between accuracy and efficiency. The small grid sizes necessary to solve shock interactions give accurate

results using second-order accurate methods. Increased complexity and extra computations with higher order methods seem to be computationally inefficient.

There are two basic fundamental descriptions of kinematic deformation in continuous media: the Lagrangian (material) and Eulerian (spatial) description. Each has its own advantages and disadvantages. Lagrangian codes provide good spatial resolution relative to the number of grid cells necessary to define the problem; free surfaces and material interfaces are well defined. Lagrangian codes also provide very proficient material constitutive modeling since they are written in material coordinates. The limitations exist where severe distortion of the grid limits the code to early times of the event. Eulerian codes have a fixed coordinate system and can handle problems involving large distortions. They do not perform as well as Lagrangian codes in the areas of grid economy, interfaces between materials, and material descriptions.

Numerical algorithms used to solve the conservation equations are not without shortcomings. Hydrocodes require specific measures to overcome singularities and phenomena such as a limiting integration time step to ensure numerical stability, and an artificial viscosity term to prevent large oscillations at shock discontinuities. The development of specific algorithms take advantage of Eulerian and Lagrangian features. Understanding the limitations and operations of the hydrocodes is imperative and can be powerful tools in research. They can provide a better understanding of the physical processes and can be used to perform analytical experiments (Anderson, 1987).



### **2.1.1 Eulerian Methods**

CTH is an Eulerian hydrocode to model large deformation, multi-material, strong shock wave physics. It uses a two-step Eulerian solution scheme. The first step is a Lagrangian step where the cells deform to follow the material motion. The mesh deforms and there is no mass flux across the cell boundaries. The second step is a remesh step where the altered cells are fitted back to the Eulerian mesh. The two step process eases the handling of multiple materials.

CTH uses finite volume approximations in place of the conservation equations. The finite volume approximations were designed to conserve mass, energy, and momentum. In cases where it is difficult or impossible to conserve these quantities, user interface allows bounding of the effects. CTH has several coordinate systems for one, two, and three-dimensional geometries. Also, computational time can be saved since not all variables are required in the remeshing step.

Due to the large amount of data produced by CTH, the most efficient means of analyzing the results of large three-dimensional calculations is with graphics. CTHER is the interactive post-processing program for CTH. User specified colors and filters assist the analyst with the sophisticated graphics.

Disadvantages of CTH include continuing development of the code and the use of different models in remapping and computing cell thermodynamics. Each model has its own assumptions, strengths and weaknesses (McGlaun et al., 1990).

### 2.1.2 Lagrangian Methods

For a Lagrangian algorithm, accuracy hinges on the interactions between slave nodes and master elements. It uses a standard finite element assembly procedure to assign normal directions for the nodes on the sliding surfaces so the outside interacting surfaces of the target and projectile can be easily identified. This identification procedure occurs regardless of any erosion on either material. The algorithm exchanges momentum between interacting nodes to preserve total momentum.

Using the framework of slave nodes and master elements, if the algorithm detects a slave node inside a master element, it brings the node back to the outside surface. This algorithm also defines the outside surface by a set of normal vectors for all master elements. A nonzero normal vector will only result on outside surfaces and provides an effective average normal to the surface. When a slave node moves to the surface of an element, it transfers its momentum to the appropriate node of the master element.

The foundation of the interaction algorithm relies on two main procedures. First, it must determine the cell locations for all slave nodes. Secondly, for each master element it must:

- (1) compute surface normal vectors,
- (2) determine cells where elements are located,
- (3) by checking all slave nodes in these cells, determine if any slave nodes are in the element, and

(4) move slave nodes back to an outside surface if it is in the element and transfer the momentum to the element nodes.

The effectiveness of this procedure relies on the use of explicit time integration. In the cases of high velocity impact, the time step must be limited so that the node interference cannot exceed more than 10-20% of a zone size during a single time step. Using a cell structure fixed in space, cells are substantially larger than elements and may include many master elements and slave nodes. This helps to quickly identify associated slave nodes and master elements when dealing with large numbers of nodes and elements in three-dimensional calculations (Belytschko et al., 1987).

### **2.1.3 ALE Methods**

Arbitrary Lagrangian-Eulerian (ALE) is a combination of methods. RHALE is an ALE shock physics code written in C++, successor to CTH. It allows different mesh types and switching of regions from Lagrangian, Eulerian, or ALE based on user input or mesh distortions. It takes advantage of the benefits from both Lagrangian and Eulerian schemes. The normal mode of operation is for calculations to proceed along in Lagrangian fashion until elements experience high distortion (the user can specify the criteria). At this point in the calculations, material can flow between elements in the highly deformed portions of the mesh to reduce distortion to acceptable levels. This procedure allows accurate treatment of contact surfaces and has less numerical divergence than purely Eulerian calculations. Large deformations can occur without the calculation failing, unlike a pure Lagrangian calculation.

RHALE can compute two-dimensional Cartesian, two-dimensional axisymmetric, and three-dimensional Cartesian geometries. It uses a volume-weighted rather than an area-weighted axisymmetric element. Using an arbitrary connectivity mesh, RHALE allows an arbitrary number of elements to share a common node. A user can specify different meshes for different regions. It uses a scalar bulk viscosity formulation which has shown to be consistently better than a variety of artificial viscosity formulations.

The major drawback of RHALE is the programming language of C++. Its developers wanted to facilitate code development and maintenance, using an object oriented program. It performs some calculations less efficiently in spite of streamlining the program in the way of heap management, reference counting, and deferred expression evaluation. Coding for matrices in C++ is inefficient but the streamlining efforts brought the CPU time within 20% of FORTRAN or C coding (Budge et al., 1993).

Another method is the Element-free Galerkin (EFG) method which offers advantages over the finite element method. It is very capable of modeling running crack problems by extending the free boundaries associated with the crack, and it is not necessary to remesh or use arbitrary Lagrangian or Eulerian formulations as in finite element methods. One disadvantage is the essential task of imposing boundary conditions in static problems and kinematic boundary conditions in dynamic problems, which are not always elementary (Lu et al., 1995).

#### 2.1.4 Particle Methods

Smooth Particle Hydrodynamics (SPH) is a gridless Lagrangian hydrodynamic computational technique for impact modeling. The formulation of the SPH equations is simplistic, even in three dimensional geometries. Lagrangian in nature, SPH is an attractive approach for problems involving complicated geometry, large voids, fractures, or disorderly flow fields. SPH uses an artificial viscosity to help model clouds of gas impacting supersonically (Monaghan, 1985). SPH differs from other techniques by using an interpolation method of summing over "kernels" associated with each particle. Each kernel is a spherically symmetric function centered at the particle location and generally Gaussian in shape.

There are two primary SPH codes at Los Alamos National Laboratory, SPHC and SPHINX. SPHC runs on several platforms in C. SPHINX is a vectorized Cray version with a clearer interface. One of the primary areas of code development for SPH codes is the installation and testing of material property routines and data bases. Current models use a custom temperature / energy relation incorporating solid / liquid / vapor / ion phases. There are plans for additional equation of state options. SPHC also incorporates numerical techniques such as variable smoothing length and particle division to model low density regions, arbitrary dimensionality and geometry, ghost particle boundary conditions, and interactive graphics.

Simulations run under SPHC and actual impact tests show similarities. Graphics produced by SPHC reveal liquid / vapor material formed from the

fractured projectile and bumper. Experimental radiographs correspond, but exact details such as the structure of the liquid / vapor phases are not exactly consistent. Developers expect details to improve with upgrades to the equation of state package (Stellingwerf et al., 1993).

SPH shows potential for future simulation of hypervelocity impacts. It can produce and track debris fragments and allow computations of secondary impacts over virtually unlimited distances. Shortfalls include the maturity of the method, since developers are still refining their codes, and mass is not conserved exactly in all problems. SPH currently uses a fragmentation model for the debris formed from a certain type of impact. However, researchers are also conducting tests on a fracture model to determine how an object disintegrates. Additionally, developers are upgrading material models (Monaghan, 1992).

Advantages of SPH include its conceptual simplicity, ease of adding new physics, natural treatment of voids, the ability to handle high strains and deformations in a Lagrangian frame, and the ability to track the debris clouds from a hypervelocity impact (Libersky et al., 1993). SPH can be used on many non-linear problems. Even in three dimensions, SPH is about 900 lines of code, making it one of the shortest and simplest of all codes (Monaghan, 1988).

In a recent simulation of brittle solids, Benz et al. (1995) used a "perfect gas" simplification. The implementation added a significant onus in complexity and computational requirements. A particle now required 17 variables: three position coordinates, three velocities, density, energy, smoothing length, mass, five components of the deviatoric stress tensor, damage and activation threshold. Even

these additions only modeled tensile failure. Although the computer time increased significantly, SPH reproduced very accurate results.

Other drawbacks of SPH include accuracy at material interfaces and problems in tension that are related to the smoothing function. Most SPH techniques use a grid stabilizing artificial viscosity that is overly simplified. SPH works well for problems with smoothly varying velocity gradients in the interior of a deforming material. However, many problems do not fall into this category (Johnson et al., 1996).

#### **2.1.5 PIC Methods**

The particle-in-cell (PIC) method models particles that are interpreted as material points on a fixed Eulerian grid. With some recent changes, the grid can also be considered an updated Lagrangian grid, thus allowing PIC to take advantage of Eulerian and Lagrangian schemes. Mesh tangling is avoided while materials are tracked through their complete deformation history.

Modifications to PIC have led to a numerical algorithm with several benefits. The updated grid eliminates the diffusion usually associated with convective derivatives in Eulerian schemes. Material points are followed to eliminate the need for interpolation of history dependent variables. There is no mesh distortion and no requirement to remesh when large deformations occur (Sulsky et al., 1994).

#### **2.1.6 Mixed Methods**

Because Lagrangian codes are highly accurate, except in cases of high distortion, incorporating an SPH algorithm into a Lagrangian code like EPIC

(Elastic Plastic Impact Computations) can produce accurate results. The SPH technique has variable nodal connectivity and can compute extreme deformations like Eulerian codes. The following steps outline how an SPH option enhances the structure of a standard Lagrangian code:

- (1) updates the velocities and displacements of the nodes (standard and SPH), based on nodal forces computed in the previous cycle,
- (2) updates the velocities and displacements of the nodes (standard and SPH) on the sliding interfaces,
- (3) determines strain rates and strains in the standard elements (using other variables to obtain the pressures, deviator stresses, and net stresses) to convert the net stresses to equivalent forces acting on the nodes, and
- (4) determines strain rates and strains in the SPH nodes by converting the nodal stresses to equivalent forces on the nodes.

SPH nodes can be coupled to a standard grid. This allows soft materials to slide and interact with stronger materials. SPH computations generally require more CPU time than standard Lagrangian calculations, so user intervention can limit SPH nodes to regions of high distortions. This can improve the efficiency of computations, by focusing the merits of SPH where they will benefit the most. When a standard grid becomes highly distorted, the SPH technique takes over and converts the highly strained elements on the interface to SPH nodes. The conversion retains the stress, strain, energy, damage, etc. of the replaced elements. The user specifies the criteria, and the code can decide where and when the SPH nodes should intervene to provide the best solution. Simulations of the coupling of



the SPH algorithm with EPIC have shown a great deal of accuracy and promise (Johnson et al., 1993).

## **2.2 PARTICLE CODES**

The Hypervelocity Microparticle Impact (HMI) project (1990) obtained data from four hydrocodes, SPH, EPIC, MESA, and CALE to understand the differences between macroscopic data and HMI data from a computational standpoint and to conduct a detailed code comparison in several areas.

Smooth Particle Hydrodynamics (SPH) is a gridless Lagrangian method using pseudo-particle interpolation to compute hydrodynamic variables. Every pseudo-particle has mass, Lagrangian velocity, and internal energy. Interpolation or constitutive relations provide the basis for other quantities. UNIX workstations and Cray computers are the primary platforms for such codes.

EPIC is a Lagrangian hydrodynamics code with the benefit of handling multi-material computations. Its primary use is in the areas of ballistic and hypervelocity impacts and jet / fragment formations. EPIC uses explicit time integration coupled with a lumped mass formulation. Advantages are the use of a cumulative damage / fracture model, erosion and total failure computations, penetration calculations without mesh distortion problems, and the use of a master-slave sliding interface to model sliding surfaces, voids, and failure. EPIC can accommodate a variety of geometric shapes and can employ the standard hydrocode boundary condition options.

MESA is an explicit Eulerian hydrocode, capable of multi-material computations, and is used primarily for armor / anti-armor applications. The heart

of the code is a second order accurate finite difference scheme with a staggered grid. During calculations, each cycle has two phases: a Lagrangian phase that updates material densities, velocities, and internal energies and an advection phase that computes the transport of mass, internal energy, and momentum. MESA can also handle a variety of geometries and has standard hydrocode boundary condition options.

CALE ("C" ALE) is a two-dimensional Arbitrary Lagrangian Eulerian hydrocode written in C. The C language gives the code the portability over a variety of computers and high flexibility in defining complex data structures. CALE can handle multi-material Eulerian flow using a volume fraction method to track interfaces. A hybrid C vectorizing compiler used on Cray computers produces highly optimized coding. CALE can run on a variety of platforms.

Although all four codes are different, testers ensured each problem was the same in the areas of initial conditions, equations of state, strength parameters, discretization, and boundary conditions. The crater depth and volume computed by all the codes were different than the experimentally observed values. MESA and SPH crater diameters were larger than the experiment while EPIC and CALE were smaller. The crater shapes were similar to the experimental crater for all the codes. By increasing the yield strength by a factor of five, the calculated craters were made smaller in diameter and depth. These calculations used a simple elastic perfectly plastic strength model, except for CALE. Another item noted was that differences appeared in crater shapes at higher velocities (greater than 20 km/s).

One cause may be the melt and vaporization phase change effects. SPH includes phase change effects whereas EPIC and MESA do not.

These tests investigated the effect of changing yield strength on impact problems. Strain-rate theory implies a change of 4.7 in yield strength should produce a change in crater volume of 3.7. The codes changed the yield strength factor by 5.0 and produced volume changes between 3.3 and 4.4, in agreement with theory. Generally, there is good agreement between codes, except that SPH and MESA agree well at low velocity but begin to disagree at intermediate velocities. More data must be extracted from these tests and the HMI data to verify their validity and further quantify their differences in predicting effects of hypervelocity impacts (Wingate et al., 1993).

### **2.3 APPLICATION OF PARTICLE CODES**

Orbital debris, much of it man-made, may restrict the exploration of space. In addition to the current population of fragments, a new concern may exacerbate the problem. "Cascading" refers to impacts between space debris in orbit that create even more fragments. Even if no more fragments enter space from earth, the total population of the space debris will continue to grow. Eventually, space exploration and travel will be perilous. Although scientists are unsure of the exact quantities of space debris and growth models, the probability that a spacecraft might be damaged by the impact of orbital debris cannot be discounted.

There have always been some particles passing near the earth capable of doing harm to a spacecraft. Asteroids, cometary debris, and not so common interstellar particles are all sources. The likelihood of an encounter with one of

these particles remains small. However, since modern space exploration began, there has been a large flux of particles, one centimeter or larger, capable of causing serious damage to spacecraft. Shields and fragmentation mitigators are the most common defense for the spacecraft. In using predictive models and data for design, there is a consensus that the greatest threat is from fragments one millimeter to ten centimeters in length. Shielding for debris less than one millimeter is possible, while weight-effective shielding for fragments larger than ten centimeters is not likely to be available for quite some time. The number of these large fragments and the probability of an encounter with one remains minute.

In 1990, an assessment of hazards to spacecraft concluded that natural hazards and operational payloads accounted for 7% of the total number of objects in orbit. On the other hand, operational debris and fragmentation debris comprised 50% and 40%, respectively. Orbital debris comes from propellants left aboard boosters that leak past seals, resulting in a reaction that fragments the spacecraft. Also, debris originates from secondary fragment and satellite collisions to create thousands to millions of particles. An estimated relative velocity impact model for space debris in orbit shows the most likely velocity is about 13 km/s for debris impacts on a spacecraft. The difficulty associated with using current launchers (6-8 km/s for a 30 gram mass) makes it imperative to extrapolate test results. To compound the problem, secondary debris fragments change in phase from solid to melt to vapor as the impact velocity increases in the range of interest. Predictions are thus difficult.

Modern launchers can achieve the appropriate velocity range, but with only the lowest masses. These launchers also launch spherical particles, while orbital fragments are usually more like plates. Experiments with flat plate impacts show more damage than with spheres of equal mass. Launchers must use irregular fragments to simulate actual space debris and different impact angles for these tests. Also, much actual testing is done on scaled models. The use of scaled models requires that fragment statistics be accurate.

There is much to learn in the area of solving the man-made space debris problem. The rate of debris propagation is slowing as international efforts are applied to mitigate the problem, but the total number of fragments continues to grow. In the near term, we must employ measures to protect the spacecraft as the risk of space exploration increases (Isbell et al., 1993).

### **2.3.1 Linking Codes**

The development and application of a Lagrangian modeling technique for debris cloud simulation is a necessity in the study of hypervelocity impacts. Many models have oversimplified their codes to accommodate difficulties in modeling debris clouds. Many computational simulations of hypervelocity impact problems use Eulerian schemes that model perforation and erosion well. However, Eulerian hydrocodes do not work well in the design of multiple plate space debris shields, where cloud debris effects occur. They have difficulty handling very low density debris (mass dispersion problems), even in two dimensions. Although there are various efforts to increase the efficiency of Eulerian schemes, they require

significant user intervention, and three dimensional analysis using Eulerian codes is impractical with current computers.

In a multi-plate shielding impact, there are regions of perforation and debris cloud evolution. Since Eulerian and Lagrangian codes each have their strengths and weaknesses, linking the codes to perform computations in their regions of suitability would be ideal. Eulerian codes are best used to predict initial debris cloud mass and velocity distribution. Lagrangian codes work best to trace stress and strain history in solid materials. There is a scheme (Fahrenthold, 1993) that links an Eulerian code, CTH, and a Lagrangian code, DYNA2D, that has produced promising results. The approach is basic. The mass and velocity distribution data from the CTH model of initial perforation is post-processed to provide a DYNA2D model of the debris cloud behind the perforated plate. The DYNA2D debris cloud model then simulates the transport of debris toward the next shield. The DYNA2D model is then post-processed to start a new CTH simulation for the perforation of the next plate. This process is repeated through the multi-plate system and finally models impact on the protected structure.

Using the scheme outlined above, a comparison of a simulation with a corresponding experiment revealed an overestimation of the energy of the debris cloud. The results did indicate that the analysis procedure provides a good, realistic transformation, from Eulerian to Lagrangian frames, of debris cloud evolution. This simulation approach for shield design applications must undergo further assessment. The combined Eulerian - Lagrangian hydrocode modeling approach shows the following:

- (1) mixture theory models for solid and fluid materials with voids can extend the use of Lagrangian hydrocodes to multidimensional debris cloud modeling, and
- (2) Lagrangian debris cloud modeling can give acceptable computation times for simulations of multi-plate impact experiments in two dimensions.

Another more recent scheme (Fahrenthold, 1995) links CTH to DYNA3D to model oblique hypervelocity impacts in three dimensions, using a particle code, DC3D. Computer time requirements of DC3D are very reasonable when compared to complete Eulerian schemes of Whipple shield impact simulation. DC3D can reduce the computation time by a factor of five. Additionally, this procedure takes into account arbitrarily nonuniform, three dimensional velocity, density, and void distributions, which are often neglected for simplification in other models. This scheme makes use of known strengths of available codes while reducing computational time for oblique hypervelocity impact problems. These improvements are necessary for computer codes to produce a practical design tool.

### **2.3.2 Multi-Bumper Designs**

There are many shielding designs that may provide adequate protection. However, testing all the different systems may be a burden, unless a particle code is used. Meteoroid velocities range from 11 to 72 km/s. Most impacts will be oblique with less than 15% within 10-20° of perpendicular. While impact tests can assist in the derivation of ballistic limit equations up to the highest laboratory velocity, analytical / numerical methods are necessary to determine shield response beyond actual test capabilities.

The Whipple shield consists of a bumper at some stand-off distance from a rear wall. In testing a Whipple shield, three regions of interest were found, based on the normal impact velocity component. At low velocities ( $<3$  km/s) impact shock pressures are low and the projectile remains basically intact after impact with a bumper. The intact but deformed projectile then impacts the shield's rear wall. As the velocity increases in the low velocity region, critical particle diameter ( $d_c$ ) decreases. In the intermediate region where velocities are greater than 3 km/s, the projectile fragments on the bumper and begins to melt above 5.5 km/s. A fragmenting or melting projectile is less damaging to the rear wall, so critical particle diameter increases in the intermediate range as velocity increases. In the high velocity region ( $>7$  km/s), the debris cloud impacting the rear wall contains solid, liquid, and vapor components of the projectile and bumper. For oblique impact angles greater than  $65^\circ$ , bumper fragments are the primary source for rear wall damage.

A major factor governing the effectiveness of Whipple shields is the condition of the debris cloud moving from the bumper toward the rear wall. Whipple shields are less effective at low impact velocities and at certain oblique angles at higher speeds. These are the conditions that produce low impact pressures in the projectile and bumper and result in solid, more penetrating fragments that impact the rear wall.

The Nextel Multi-Shock (MS) Shield is a low weight variation to the Whipple shield. It consists of four equally spaced ceramic fiber bumpers with an aluminum rear wall. Weight savings occur, in reducing the rear wall thickness to



stop a given particle. Oblique impacts do not produce damaging fragments with the ceramic fiber bumpers. Particles from the fabrics are short fibers which generally do not penetrate subsequent layers and do not damage the rear wall. These fibers are lighter than Whipple shield fragments and are stopped by lower layers of the MS shield.

The Mesh Double Bumper (MDB) Shield is another system, similar to the MS shield. It is simply a Whipple shield geometry with two new layers. The first is a mesh bumper in front of the Whipple shield, and the second is a high strength fabric layer between the Whipple shield and rear wall. Tests have shown above average performance for this double bumper system with a mesh. The mesh is a mass efficient method to disrupt the projectile and spread the debris cloud.

Multi-bumper shields have several advantages over Whipple shields, besides improved performance. Multi-bumper designs produce less damaging secondary ejecta in oblique impacts. They produce a series of shocks on a projectile that increases the heating of the projectile, turning the kinetic energy into internal thermal energy. They are less sensitive to projectile shape because the multiple shocks disrupt the projectile to a greater extent. Multi-bumper systems cause greater projectile fragmentation and slow the expansion of the debris cloud. Finally, because smaller and fewer particles impact the rear wall, multi-bumper shields produce less cumulative damage over time (Christiansen, 1993).

## **2.4 SUMMARY**

There are many shielding designs that may provide adequate protection. However, testing all the different systems may be a burden, unless a particle code

is used. Meteoroid velocities range from 11 to 72 km/s, well beyond the capabilities of light gas guns. Most impacts will be oblique, with less than 15% within 10-20° of perpendicular. While impact tests and experiments can assist in the derivation of ballistic limit equations up to the highest laboratory velocity, interpolation must be used beyond actual test capabilities. Analytical / numerical methods become imperative to determine shield response, given experimental limitations.

There are many codes and methods to simulate hypervelocity impact problems, but none that can solve every variation of a hypervelocity impact problem. However, each model has its own strengths and weaknesses, using basic assumptions and properties. Even with current supercomputers, modeling hypervelocity impact events can be taxing in just two dimensions. There is a zone of acceptability that rests on acceptable CPU times and accurate modeling and results. More testing and verification of current codes is necessary to refine the modeling methodology. Additionally, more computer development or streamlining of codes can reduce CPU time. These developments will assist in designing shield systems for hypervelocity impact protection.

## Chapter 3: Whipple Shields

### 3.1 WHIPPLE SHIELD BALLISTIC LIMIT EQUATIONS

The Whipple shield is the basis for nearly all simple, orbital debris shielding designs (Figure 1). The Whipple shield is a sacrificial, thin sheet of material, or bumper, that is at a predetermined standoff distance from a rear wall. An impacting projectile hits the bumper, fragments, and is dispersed. The bumper transforms the projectile's kinetic energy into thermal energy. The resulting debris cloud of projectile and bumper fragments is less concentrated than the original projectile when it reaches the rear wall.

The equations in this section define the ballistic limit curve of an aluminum alloy, single Whipple shield. The critical particle diameter ( $d_c$ ) is characterized as causing shield failure by either complete penetration or detached spall. Experimental tests have shown three distinct regions based on the normal impact velocity component of the projectile. The first region is below 3 km/s. At these low velocities, the projectile remains nearly intact after the impact with the bumper. Shock pressures are low, but the rear wall sustains impact from a deformed but sound projectile. As velocity increases in the low velocity region, the projectile is more damaging. The intermediate region is between 3 and 7 km/s. At normal velocities above 3 km/s, the projectile breaks apart on the bumper and begins to melt at higher velocities. This fragmenting and partially molten projectile is less damaging to the rear wall in this second region. The third region consists of normal velocities above 7 km/s, where the debris cloud that impacts the rear wall

contains all phases of projectile and bumper components. The debris cloud is more damaging as velocity increases in this region (Christiansen, 1993).

The Hypervelocity Impact Technology Facility defines the protection capability threshold for a Whipple shield in terms of a particle diameter ( $d_c$ ) that causes shield failure. Failure is defined as perforation or spall of the rear wall. The three regions are delineated based on the normal impact velocity component ( $V_n$ ). Christiansen's (1993) ballistic limit curve is given by the following equations:

For  $V_n \leq 3$  km/sec:

$$d_c = \left[ \left( t_w \left( \frac{\sigma}{40} \right)^{0.5} + t_b \right) / \left( 0.6 (\cos \theta)^{5/3} \rho_p^{0.5} V^{2/3} \right) \right]^{(18/19)} \quad (3-1)$$

For  $3 \text{ km/sec} < V_n < 7 \text{ km/sec}$ :

$$d_c = \left\{ \left[ \left( t_w \left( \frac{\sigma}{40} \right)^{0.5} + t_b \right) / \left( 1.248 \rho_p^{0.5} \cos \theta \right) \right]^{(18/19)} \times \left( 1.75 - \frac{(V \cos \theta)}{4} \right) \right\} + \left\{ \left[ 1.071 t_w^{2/3} \rho_p^{-1/3} \rho_b^{-1/9} S^{1/3} \left( \frac{\sigma}{70} \right)^{1/3} \right] \times \left( \frac{(V \cos \theta)}{4} - 0.75 \right) \right\} \quad (3-2)$$

For  $V_n \geq 7$  km/sec:

$$d_c = 3.918 t_w^{2/3} \rho_p^{-1/3} \rho_b^{-1/9} (V \cos \theta)^{-2/3} S^{1/3} \left( \frac{\sigma}{70} \right)^{1/3} \quad (3-3)$$

where the terms are defined as

- $\theta$  angle of impact from surface normal (deg)
- $\sigma$  rear wall yield stress (ksi)

$\rho_b$	density of bumper ( $\text{g/cm}^3$ )
$\rho_p$	density of projectile ( $\text{g/cm}^3$ )
$d_c$	critical diameter of projectile (cm)
$S$	spacing (cm)
$t_b$	thickness of bumper (cm)
$t_w$	thickness of rear wall (cm)
$V$	velocity of projectile (km/sec)

### 3.2 METHODOLOGY

Two matrices of 10 simulations each were developed to evaluate EXOS impact simulations at 15° and 45°, respectively. Initial estimates of the ballistic limit were obtained from the equations above. Impact simulations were conducted for projectile diameters slightly smaller and slightly larger than this limit at velocities of 3, 5, 7, 9, and 11 km/s. Tables 1 and 2 show that for each configuration and impact velocity, the projectile size was varied until the ballistic limit was determined by simulation. The most current version of the code was used and is annotated in the tables. For each velocity, EXOS was run to different stop times, usually 6-15 times the time required for an unimpeded projectile of the same initial velocity to traverse the space between the shield and rear wall. This allowed sufficient time for the debris cloud to impact the rear wall and generate an impulse load. In all cases, the debris cloud was examined to determine the potential for further rear wall damage. Table 3 shows the material properties used in the impact simulations.

The geometry under consideration contains a 0.127 centimeter thick AL 6061-T6 bumper spaced 5.08 centimeters from a 0.3175 centimeter thick AL 6061-T6 rear wall. The impact angles were 15° and 45°. Velocities ranged from 3 - 11 km/s, and projectile diameters ranged from 0.20 - 0.50 centimeters. These simulations demonstrate the accuracy of the results. Tables 4 and 5 summarize the final results of the 15° and 45° impact simulations. In cases where the simulation did not initially match the experimentally derived ballistic limit curve, the simulation was rerun at a higher resolution (more particles), using the most current version of the code.

### **3.3 COMPARISON OF EXOS SIMULATIONS TO THE BALLISTIC LIMIT CURVE**

Figures 2 and 3 show a set of ballistic limit curves for a 5.08 centimeter standoff Whipple Shield with 15° and 45° impact obliquities, respectively. Results of the EXOS simulations are plotted with the curves. The curves show the critical particle diameter that fails the shield, as a function of projectile velocity. Failure, defined as perforation or spall, is above the curve. Figures 4 and 5 show an EXOS result for no failure at 15°. Figures 6 and 7 show an EXOS result for failure at 15°. Figures 8 and 9 show no failure at 45°. Figures 10 and 11 show failure at 45°.

For the 15° simulations, EXOS accurately predicts the critical particle diameter in the low and intermediate velocity regions. In the 45° simulations, EXOS predicts the critical particle diameter in all three velocity regions. For example, in the 45° case at 7 km/s, the ballistic limit equations predict failure for a 0.40 centimeter diameter projectile. EXOS predicts failure for the same velocity

between 0.35 and 0.45 centimeters, closely matching the experimental results. In the high velocity region, EXOS closely matched the experimentally derived ballistic limit curve, accurately predicting Whipple shield performance in the 45° case, but overpredicting damage in the 15° case at 11 km/s. All simulation results are represented in Figures 2 and 3. Inspections of the 45° results (Figures 8-11) show that much of the projectile fragments are displaced laterally, never reaching or penetrating the rear wall.

One explanation for the overestimation of damage to the rear wall at high velocity is in the modeling of the shielding geometry. The Whipple shield simulations were conducted with a coarse mesh from 6,600 to nearly 36,000 particles for each simulation. The impacting projectile contained two particles across the sphere radius. This somewhat coarse mesh concentrated the kinetic energy in fewer particles. The resulting debris cloud consisted of large fragments, rather than a fine mist. This concentration of momentum would create more damage than if the energy were distributed through more particles in the debris cloud.

Also, some of these simulations were run with EXOS version 20.02. Version 20.02 assumes a constant yield strength and shear modulus. It does not include "thermal softening" of materials, where the yield strength is a function of temperature. The dependence of yield strength on temperature will influence the degree of terminal damage. A newer version of the code, 20.23, that includes "thermal softening" and improved viscous effects, shows more accurate results with coarse mesh simulations. Version 20.23 was made available during the

course of these test simulations. However, it was used on simulations that required an increase in particle density. Also, the simulations were terminated well beyond the point when the initial debris cloud of projectile and bumper particles impacted the rear wall. However, the rear wall could receive subsequent loading from the remaining debris cloud and a late time structural failure from the impulse of the debris impact. CPU constraints dictated the termination of the simulation after a reasonable time.

The sensitivity of the simulations to the experimental ballistic limit curve can be illustrated with the following example: a small (25%) increase in particle diameter (from 0.40 centimeters to 0.50 centimeters) can increase the volume and mass accordingly of the projectile by 95%. This one millimeter increase in particle diameter alone nearly doubles the kinetic energy of the projectile. Although the scale of orbital debris shielding design is small, the consequence of projectile variation can be immense. From this observation, it is easier to appreciate the sensitivity of the ballistic limit curve.

Tables 6 and 7 show the CPU time and memory requirements for each impact simulation. The rezoning option was used on all simulations, at 2.75 CPU hour increments. Rezoning used very little CPU time, but by eliminating particles that migrated outside of the specified geometry, it allowed the analysis code to minimize the number of overall cells, focusing on the particles and rear wall interaction. Simply doubling the number of cells spanning each of the three dimensions of a plate increases the zoned region in the overall model nearly eight times. Rezoning has a higher payoff as the number of particles increases and at



later simulation times, as some of the debris propagates away from the rear wall. An increase of model complexity can be seen in the CPU requirements for simulations in Tables 6 and 7, with more than 35,000 particles.

## **Chapter 4: Stuffed Whipple Shields**

### **4.1 STUFFED WHIPPLE SHIELD BALLISTIC LIMIT EQUATIONS**

Another low weight shield system designed to improve spacecraft protection from orbital debris is the stuffed Whipple shield. The stuffed Whipple shield can improve the protection performance of a Whipple shield with small distances between the shield, or bumper, and rear wall. It is basically an enhanced Whipple shield with an intermediate layer of material between the rear wall and the aluminum bumper. This material is usually a combination of Nextel and Kevlar (Figure 4). Alternative stuffed Whipple designs use aluminum, Kevlar, or Nextel alone as the intermediate bumper. The intermediate layer generates high shock pressure and disrupts the impacting projectile and debris cloud, making them less lethal when they reach the rear wall (Christiansen et al., 1995). Although the additional shield contributes particles to the debris cloud, these particles are much smaller than the fragments of the projectile. Additionally, the benefit of the intermediate shield defeating the impacting projectile outweighs its disadvantage of debris cloud contribution.

The equations in this section, from hypervelocity impact test data and analysis, define three distinct regions for the ballistic limit of the stuffed Whipple shield. The critical particle diameter ( $d_c$ ) is determined as causing shield failure by either complete penetration or detached spall. These general equations define the maximum projectile size that a stuffed Whipple shield can protect against, as a function of impact velocity and other configuration and material properties

(Christiansen et al., 1997). In the low velocity region, the projectile remains nearly intact but deformed when it impacts the rear wall, making it more damaging at higher velocities in the low region. In the intermediate velocity region, the projectile breaks apart when it impacts the bumpers and begins to melt, making it less harmful as velocity increases. The bumpers transform more of the kinetic energy into thermal energy. In the high velocity region, the debris cloud of projectile and bumper fragments is a combination of solid, melt, and vapor when it impacts the rear wall. As velocity increases in the high velocity region, the debris cloud is more damaging (Christiansen et al., 1995).

The Hypervelocity Impact Technology Facility defines the protection capability threshold for a stuffed Whipple shield in terms of a particle diameter ( $d_c$ ) that causes shield failure. The equations were derived from experimental data for a particular shield configuration given in Figure 12, and the equations are only valid for the specific geometry. The three regions are delineated based on the projectile velocity and angle of impact from the surface normal. Christiansen et al. (1997) defines the ballistic limit curve by the following equations:

For low velocity,  $V \leq 2.7 / (\cos\theta)^{0.5}$  :

$$d_c = 2 \left[ t_w \left( \frac{\sigma}{40} \right)^{0.5} + 0.37 m_b \right] / \left[ (\cos\theta)^{5/3} \rho_p^{0.5} V^{2/3} \right] \quad (4-1)$$

For intermediate velocity,  $2.7/(\cos\theta)^{0.5} < V < 6.5 / (\cos\theta)^{1/3}$  :

$$d_c = 0.321(t_w \rho_w)^{1/3} \rho_p^{-1/3} (\cos\theta)^{-7/18} S^{2/3} \left(\frac{\sigma}{40}\right)^{1/6} \times$$

$$\left[ \frac{\left(V - \frac{2.7}{(\cos\theta)^{0.5}}\right)}{\left(\frac{6.5}{(\cos\theta)^{1/3}} - \frac{2.7}{(\cos\theta)^{0.5}}\right)} \right] + 1.031 \rho_p^{-1/2} \left[ t_w \left(\frac{\sigma}{40}\right)^{0.5} + 0.37 m_b \right] \times$$

$$(\cos\theta)^{-4/3} \left[ \frac{\left(\frac{6.5}{(\cos\theta)^{1/3}} - V\right)}{\left(\frac{6.5}{(\cos\theta)^{1/3}} - \frac{2.7}{(\cos\theta)^{0.5}}\right)} \right] \quad (4-2)$$

For high velocity,  $V \geq 6.5 / (\cos\theta)^{1/3}$  :

$$d_c = 0.6(t_w \rho_w)^{1/3} \rho_p^{-1/3} V^{-1/3} (\cos\theta)^{-0.5} S^{2/3} \left(\frac{\sigma}{40}\right)^{1/6} \quad (4-3)$$

where the terms are defined as

- $\theta$  angle of impact from surface normal (deg)
- $\sigma$  rear wall yield stress (ksi)
- $\rho_b$  density of bumper ( $\text{g/cm}^3$ )
- $\rho_p$  density of projectile ( $\text{g/cm}^3$ )
- $d_c$  critical diameter of projectile (cm)
- $m_b$  areal density of all bumpers ( $\text{g/cm}^2$ )
- $S$  spacing (cm)

- $t_w$       thickness of rear wall (cm)  
 $V$       velocity of projectile (km/s).

## 4.2 METHODOLOGY

The equations in this chapter are for a particular geometry, and when applied to an all aluminum configuration, are non-conservative in some cases. The equations will be modified after further hypervelocity experiments and analyses are completed at the Johnson Space Center (Christiansen et al., 1997). The general equations for the stuffed Whipple shield configuration given above were validated from experimental data analysis at a 11.43 centimeter standoff between the first bumper and rear wall (Figure 12) for impact obliquities of  $0^\circ$  and  $45^\circ$ .

The comparison that was conducted was between an all aluminum, stuffed Whipple configuration from experiment and an equivalent Multi-Layer Insulation (MLI), Nextel, and Kevlar stuffed Whipple geometry. The thicknesses of the bumpers for the all aluminum case (Figure 13) was determined to yield an areal density nearly equivalent to the MLI, Nextel, and Kevlar combination. The areal density of the aluminum bumper, MLI, Nextel, and Kevlar was determined to be  $1.378 \text{ g/cm}^2$ . The density of Al 6061-T6 is  $2.703 \text{ g/cm}^3$ . For the all aluminum case, thicknesses of 0.16 cm for the first bumper and 0.32 cm for the second bumper were used in experiments. This results in  $1.297 \text{ g/cm}^2$  for the areal density of the aluminum bumpers. Next, an impact angle of  $15^\circ$  from the experiment was applied to the equations for the given geometry.

A matrix of 3 simulations was used to evaluate EXOS impact simulations on a stuffed Whipple shield geometry. Initial estimates of the ballistic limit were

obtained from the equations above. Impact simulations were conducted for projectile diameters and velocities similar to those in experiments conducted by NASA Johnson Space Center. Table 8 shows each configuration, impact velocity, projectile size, and version of the code used for the EXOS simulation. For each velocity, EXOS was run to different stop times, usually four times the time required for an unimpeded projectile with the same initial velocity to traverse the space between the first shield and rear wall. This allowed sufficient time for the debris cloud to impact the rear wall and generate an impulse load. In all three cases, the rear wall failed, and the simulation was terminated. Table 9 shows the material properties used in the stuffed Whipple impact simulations.

The geometry under consideration contains a 0.16 centimeter thick AL 6061-T6 bumper spaced 11.43 centimeters from a 0.48 centimeter thick AL 2219T87 rear wall. An intermediate bumper of AL 6061-T6 is placed midway between the first bumper and rear wall, at 5.715 centimeters from each. The impact angle was  $15^\circ$ . Velocities ranged from 6.64 - 6.81 km/s, and projectile diameters ranged from 0.95 - 1.0 centimeters. These simulations demonstrate the accuracy of the results. Table 10 summarizes the final results of the  $15^\circ$  stuffed Whipple impact simulations.

#### **4.3 COMPARISON OF EXOS SIMULATIONS TO THE BALLISTIC LIMIT CURVE**

Figure 14 shows the ballistic limit curve for a Nextel / Kevlar stuffed Whipple shield with a 11.43 centimeter standoff, intermediate bumper at midway, and  $15^\circ$  impact obliquity. Results of the EXOS simulations are plotted with the curve. The curve is of critical particle diameter that fails the shield as a function of

projectile velocity. Failure, defined as perforation or spall, is above the curve. Figures 16 and 17 show an EXOS all aluminum, stuffed Whipple result, with wall failure.

For the all aluminum, stuffed Whipple simulations, EXOS predicted failure for all three simulations. Experimental results reveal shield failure only for simulations SW6 and SW8; EXOS overpredicted damage for simulation SW7. All simulation results are represented in Figure 14. Inspection of an all aluminum stuffed Whipple result (Figures 15-16) show that much of the projectile fragments and debris cloud have reached the rear wall.

One explanation for the overestimation of damage to the rear wall is in the modeling of the shielding geometry. The stuffed Whipple shield simulations were conducted with a coarse mesh of less than 35,000 particles for each simulation. The impacting projectile contained three particles across the sphere radius. This somewhat coarse mesh concentrated the kinetic energy in fewer particles. The resulting debris cloud consisted of large fragments, rather than a fine mist. This concentration of momentum would create more damage than if the energy were distributed through more particles in the debris cloud.

Although the simulations were terminated after the first indication of failure (spallation), the debris cloud of projectile and bumper particles was still impacting the rear wall. The rear wall could receive subsequent loading from the remaining debris cloud and further structural failure from the impulse of the debris impact. CPU constraints dictated the termination of the simulation after a reasonable time.

These results are consistent with Johnson Space Center data (Christiansen et al., 1995) which found that Nextel and Kevlar stuffed Whipple shields provide better protection than all aluminum, stuffed Whipple shields. Nextel and Kevlar bumpers compared to all aluminum weight equivalents can defeat 50-300% more massive projectiles. A comparison (Figure 14) of one of the EXOS simulations at ~6.8 km/s and 1.00 centimeter shows a Nextel / Kevlar configuration defeating a projectile 1.4 centimeters, or 2.75 times more massive.

Table 11 shows the CPU time and memory requirements for each impact of the all aluminum, stuffed Whipple simulations. The rezoning option was used on all simulations, at 2.75 CPU hour increments. Again, rezoning used very little CPU time, but by eliminating the particles that migrated away from the rear wall, it allowed the code to minimize the overall modeled geometry and focus on the debris cloud and rear wall interaction.



## Chapter 5: Conclusions

A new particle-finite element code, EXOS, was used to simulate hypervelocity impacts in the application of orbital debris shielding design. A comparison of the simulations was performed on a series of single and multi-plate aluminum shielding geometries. This code was evaluated by determining the ballistic limit of a Whipple shield at velocities between 3 and 11 km/s, requiring a total of 20 simulations. The simulation process was fully documented and data was compared to ballistic limit curves derived from experimental derivations. The evaluation shows that EXOS agrees well with experiment in the low and intermediate velocity regions ( $< 7$  km/s), but the code tends to overestimate damage at high velocity. EXOS was able to produce reasonable results in the untestable region. Gaining confidence in the code, the evaluation was extended by comparing code simulations to experimental results of an all aluminum, stuffed Whipple geometry and to the ballistic limit curve of a Nextel / Kevlar stuffed Whipple shield. Again, the stuffed Whipple shield ballistic limit curve was derived from experimental analysis.

The information in this thesis shows that simulations from EXOS are accurate and can be accomplished in a reasonable amount of CPU time. Given the limited validation of EXOS simulations for the Whipple and stuffed Whipple geometries, the results are in general agreement with the extrapolated experimental data. More improvements to EXOS are expected, which should improve its ability to model hypervelocity impact phenomena. The average run time for the Whipple

shield simulations was 23.04 CPU hours on a Cray J90. These results support the trend towards increased use of computer simulation in orbital debris shield design.

Table 1: Projectile Diameters and Velocities Used for the 15° Impact Whipple Shield Simulations

Simulation Number	Projectile Diameter (cm)	Velocity (km/s)	Stop Time (μsec)	EXOS Version
BL3	0.200	3	100.0	20.02
BL4c	0.300	3	94.0	20.23
BL5	0.300	5	75.0	20.02
BL6c	0.400	5	70.3	20.23
BL7	0.400	7	75.0	20.02
BL8	0.500	7	75.0	20.02
BL9	0.350	9	75.0	20.02
BL10	0.450	9	75.0	20.02
BL11c	0.300	11	17.2	20.23
BL12	0.400	11	75.0	20.02

**Table 2: Projectile Diameters and Velocities Used for the 45° Impact Whipple Shield Simulations**

Simulation Number	Projectile Diameter (cm)	Velocity (km/s)	Stop Time (μsec)	EXOS Version
BLO3	0.350	3	100.0	20.02
BLO4c	0.450	3	92.6	20.23
BLO5	0.300	5	100.0	20.02
BLO6c	0.400	5	74.4	20.23
BLO7	0.350	7	90.0	20.02
BLO8c	0.450	7	67.6	20.23
BLO9	0.400	9	90.0	20.02
BLO10	0.500	9	90.0	20.02
BLO11	0.375	11	75.0	20.02
BLO12	0.425	11	75.0	20.02

Table 3: Material Properties Used in the Whipple Shield Impact Simulations

Material type: Al 6061-T6

Equation of State: Mie-Grüneisen

Value	Units	Definition
$Y = 2.9\text{e-}03$	Mbar	yield strength
$c_v = 0.885\text{e-}05$	Mbar-cm <sup>3</sup> /g-K	specific heat
$\rho_0 = 2.703$	g/cm <sup>3</sup>	reference density
$G_0 = 0.276$	Mbar	shear modulus
$Y_{\text{max}} = 5.8\text{e-}03$	Mbar	maximum yield stress
$\epsilon_f = 3.0$	-	failure strain
$\Sigma = 1.2\text{e-}01$	Mbar	fracture pressure
$C_0 = 0.524$	cm/ $\mu$ s	reference sound speed
$S_1 = 1.40$	-	Hugoniot slope
$\gamma_0 = 1.97$	-	Grüneisen's gamma

Table 4: Damage Predicted by the 15° Whipple Shield Simulations

Simulation Number	Shield Failure	Rear Wall Damage Description
BL3	No	Slight bulge (1 mm deep)
BL4c	Yes	Spallation
BL5	No	Slight bulge (1 mm deep)
BL6c	Yes	Spallation; Bulge (2 mm deep)
BL7	No	Slight bulge (2 mm deep)
BL8	Yes	Perforation; Hole (17 mm diam)
BL9	No	Slight bulge (2 mm deep)
BL10	Yes	Spallation; Bulge (25 mm diam, 9 mm deep)
BL11c	Yes	Perforation; Hole(5 mm diam)
BL12	Yes	Spallation; Bulge (12 mm diam, 2 mm deep)

Table 5: Damage Predicted by the 45° Whipple Shield Simulations

Simulation Number	Shield Failure	Rear Wall Damage Description
BLO3	No	Slight bulge (1 mm deep)
BLO4c	Yes	Spallation
BLO5	No	Slight bulge (2 mm deep)
BLO6c	Yes	Spallation
BLO7	No	Slight bulge (2 mm deep)
BLO8c	Yes	Perforation; Bulge (2 mm deep)
BLO9	No	Slight bulge (1 mm deep)
BLO10	Yes	Spallation; Bulge (3 mm deep)
BLO11	No	Slight bulge (1 mm deep)
BLO12	Yes	Spallation; Bulge (18 mm diam, 3 mm deep)

Table 6: Impact Simulation Time on the Cray J90 for the 15° Simulations

Simulation Number	Number of Particles	EXOS CPU Time (h) (<20.05 Mword)	Rezone CPU Time (h)	Total Simulation Time (h)
BL3	6550	14.293	0.049	14.342
BL4c	35,950	66.435	0.821	67.256
BL5	6505	13.056	0.051	13.107
BL6c	5226	21.531	0.039	21.570
BL7	6046	9.522	0.043	9.565
BL8	6046	6.338	0.049	6.387
BL9	6550	20.510	0.053	20.563
BL10	6550	16.330	0.051	16.381
BL11c	35,950	22.374	0.039	22.413
BL12	6550	18.758	0.047	18.805



Table 7: Impact Simulation Time on the Cray J90 for the 45° Simulations

Simulation Number	Number of Particles	EXOS CPU Time (h) ( $<20.05$ Mword)	Rezone CPU Time (h)	Total Simulation Time (h)
BLO3	6550	17.138	0.047	17.185
BLO4c	36,099	66.431	0.817	67.248
BLO5	6505	16.499	0.049	16.548
BLO6c	36,099	66.414	0.834	67.248
BLO7	6046	18.543	0.043	18.586
BLO8c	5226	21.690	0.039	21.729
BLO9	6550	19.221	0.047	19.268
BLO10	6550	17.513	0.047	17.560
BLO11	6550	18.213	0.049	18.262
BLO12	6550	16.696	0.047	16.743

**Table 8: Projectile Diameters and Velocities Used for the 1° Impact Stuffed Whipple Shield Simulations**

Simulation Number	Projectile Diameter (cm)	Velocity (km/s)	Stop Time (μsec)	EXOS Version
SW6	0.95	6.64	65.2	20.23
SW7	0.95	6.78	62.9	20.23
SW8	1.00	6.81	61.7	20.23

Table 9: Material Properties Used in the Stuffed Whipple Shield Impact Simulations

Equation of State: Mie-Grüneisen			
Material	Value	Units	Definition
Al 6061-T6	$Y = 2.9\text{e-}03$	M bar	yield strength
	$c_v = 0.885\text{e-}05$	M bar-cm <sup>3</sup> /g-K	specific heat
	$\rho_0 = 2.703$	g/cm <sup>3</sup>	reference density
	$G_0 = 0.276$	M bar	shear modulus
	$Y_{\max} = 5.8\text{e-}03$	M bar	maximum yield stress
	$\epsilon_f = 3.0$	-	failure strain
	$\Sigma = 1.2\text{e-}01$	M bar	fracture pressure
	$C_0 = 0.524$	cm/ $\mu$ s	reference sound speed
	$S_1 = 1.40$	-	Hugoniot slope
	$\gamma_0 = 1.97$	-	Grüneisen's gamma
Al 2219T87	$Y = 3.9\text{e-}03$	M bar	yield strength
	$c_v = 0.863\text{e-}05$	M bar-cm <sup>3</sup> /g-K	specific heat
	$\rho_0 = 2.785$	g/cm <sup>3</sup>	reference density
	$G_0 = 0.286$	M bar	shear modulus
	$Y_{\max} = 7.6\text{e-}03$	M bar	maximum yield stress
	$\epsilon_f = 3.0$	-	failure strain
	$\Sigma = 1.2\text{e-}01$	M bar	fracture pressure
	$C_0 = 0.5328$	cm/ $\mu$ s	reference sound speed
	$S_1 = 1.338$	-	Hugoniot slope
	$\gamma_0 = 2.00$	-	Grüneisen's gamma

**Table 10: Damage Predicted by the Stuffed Whipple Simulations**

<b>Simulation Number</b>	<b>Shield Failure</b>	<b>Rear Wall Damage Description</b>
SW6	Yes	Spallation; Slight bulge (4mm deep)
SW7	Yes	Spallation; Slight bulge (4mm deep)
SW8	Yes	Spallation; Slight bulge (5mm deep)

**Table 11: Impact Simulation Time on the Cray J90 for the Stuffed Whipple Simulations**

Simulation Number	EXOS CPU Time (h) ( $<20.05$ Mword)	Rezone CPU Time (h)	Total Simulation Time (h)
SW6	22.134	0.266	22.400
SW7	22.141	0.272	22.413
SW8	22.147	0.264	22.411

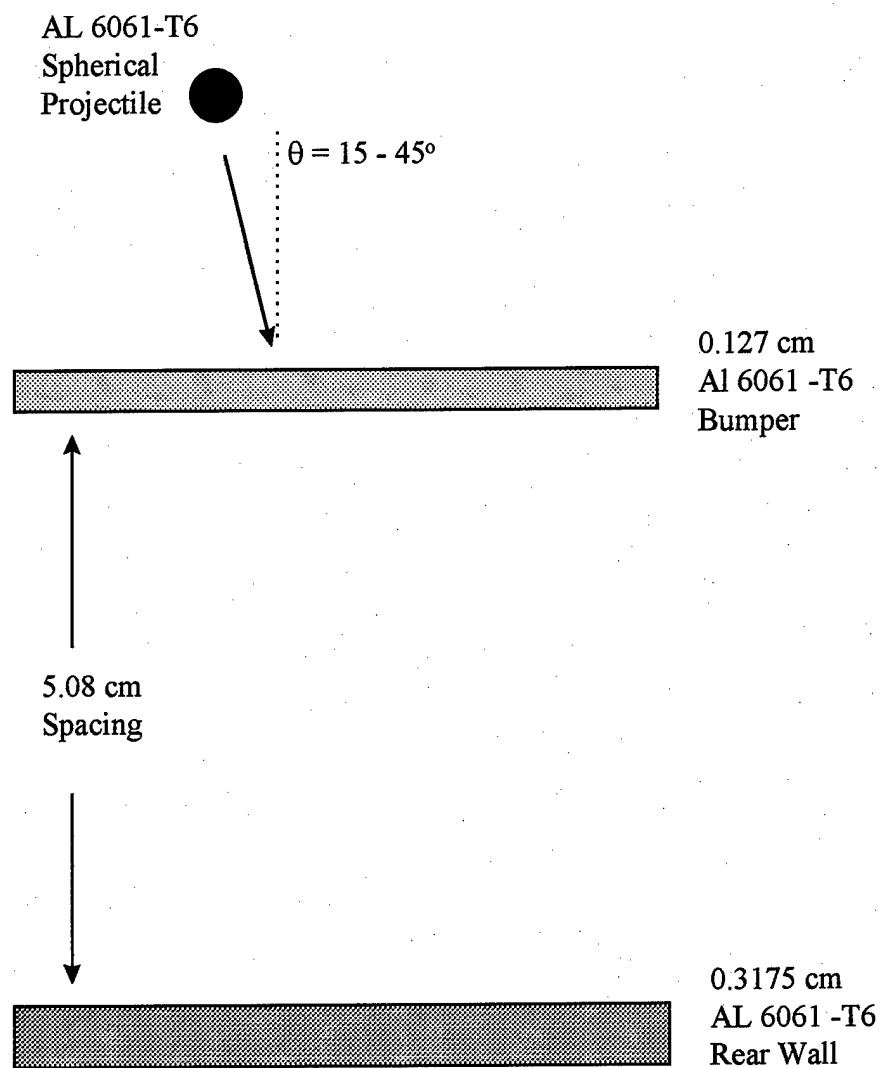


Figure 1: Whipple Shield Geometry

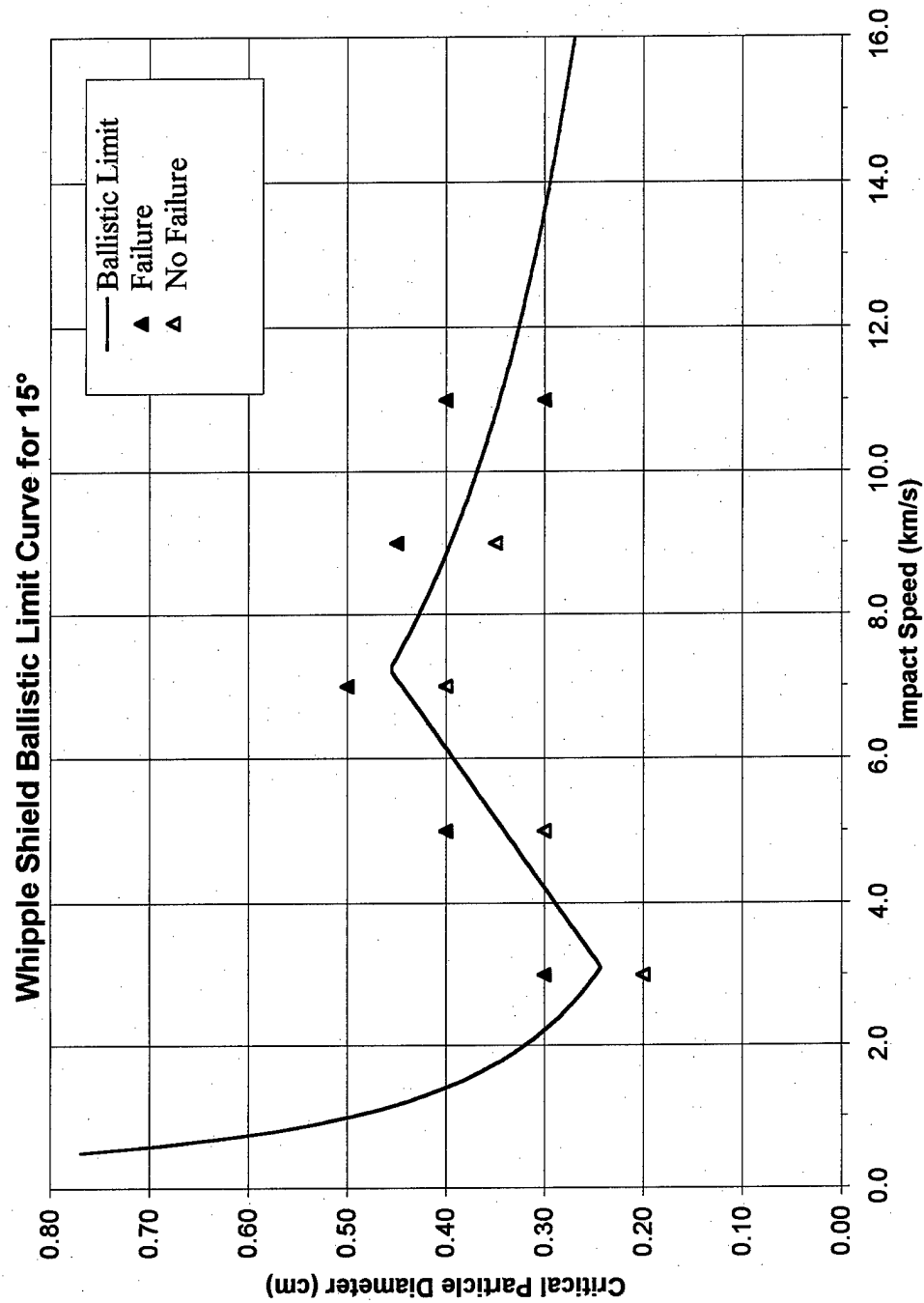


Figure 2: Ballistic Limit Curve for the 15° Whipple Shield Simulations

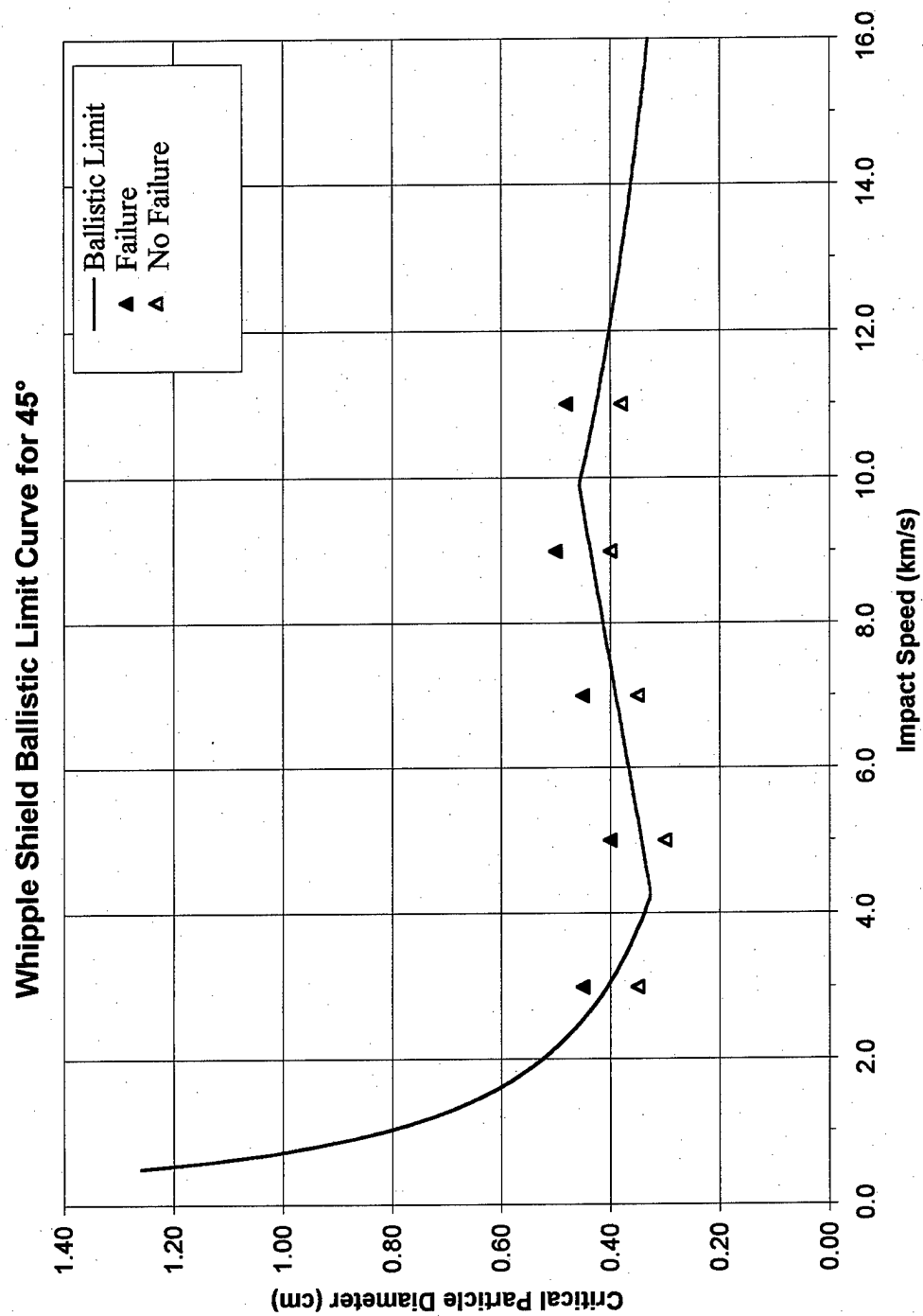


Figure 3: Ballistic Limit Curve for the 45° Whipple Shield Simulations



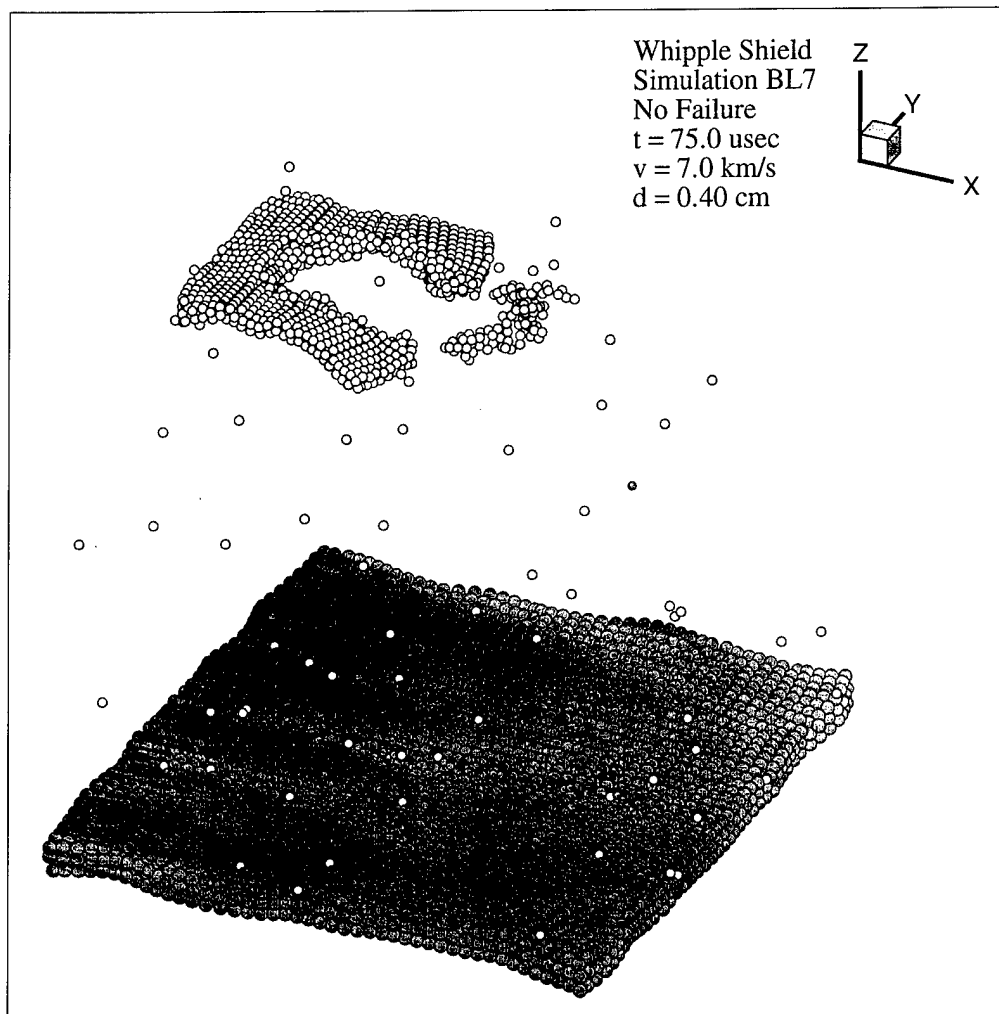


Figure 4: 15° EXOS Result for Simulation BL7

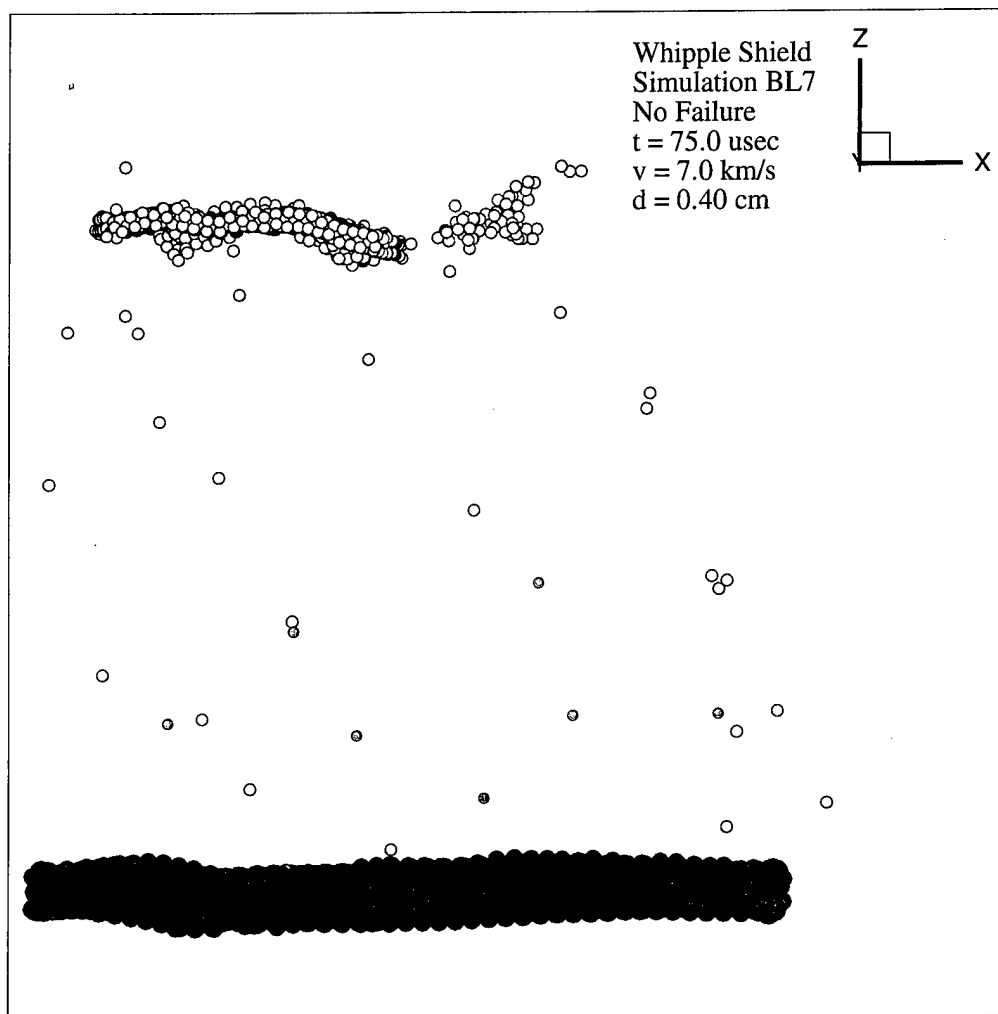


Figure 5: Side View of 15° EXOS Result for Simulation BL7

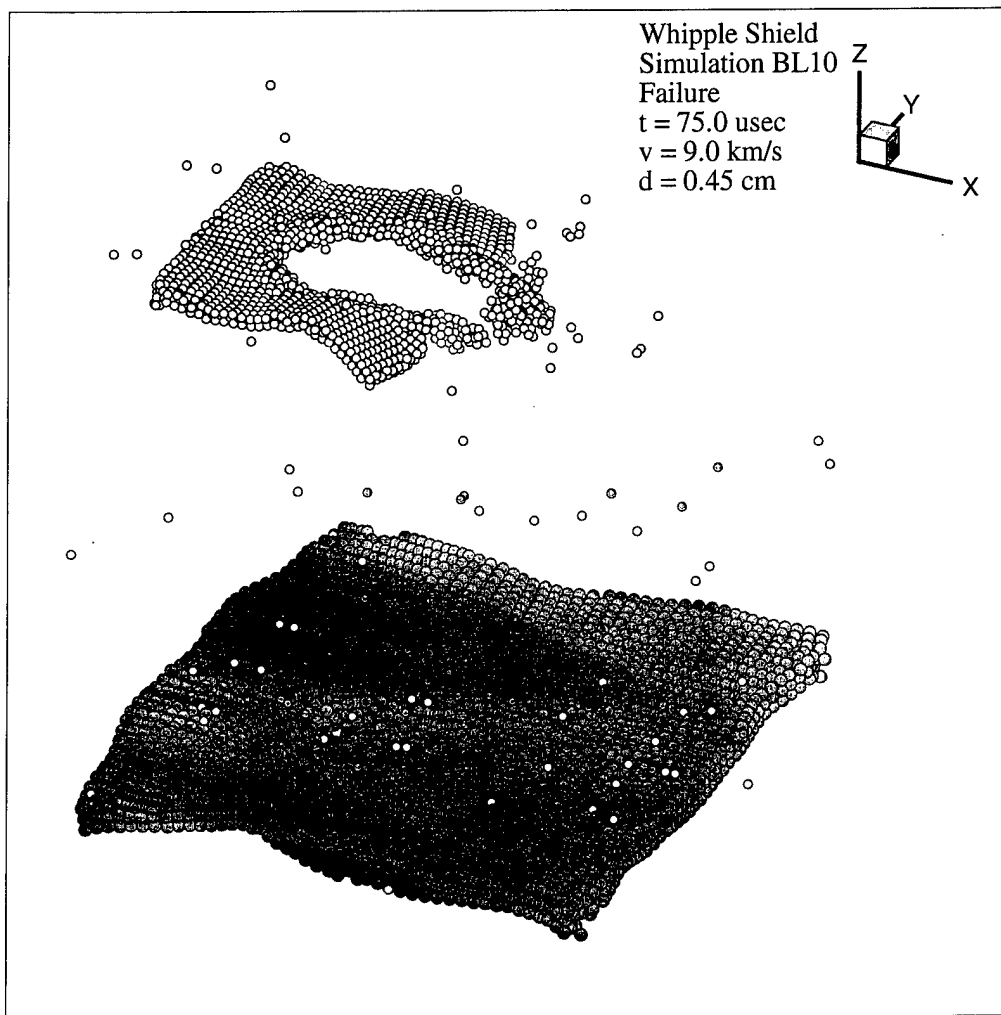


Figure 6: 15° EXOS Result for Simulation BL10

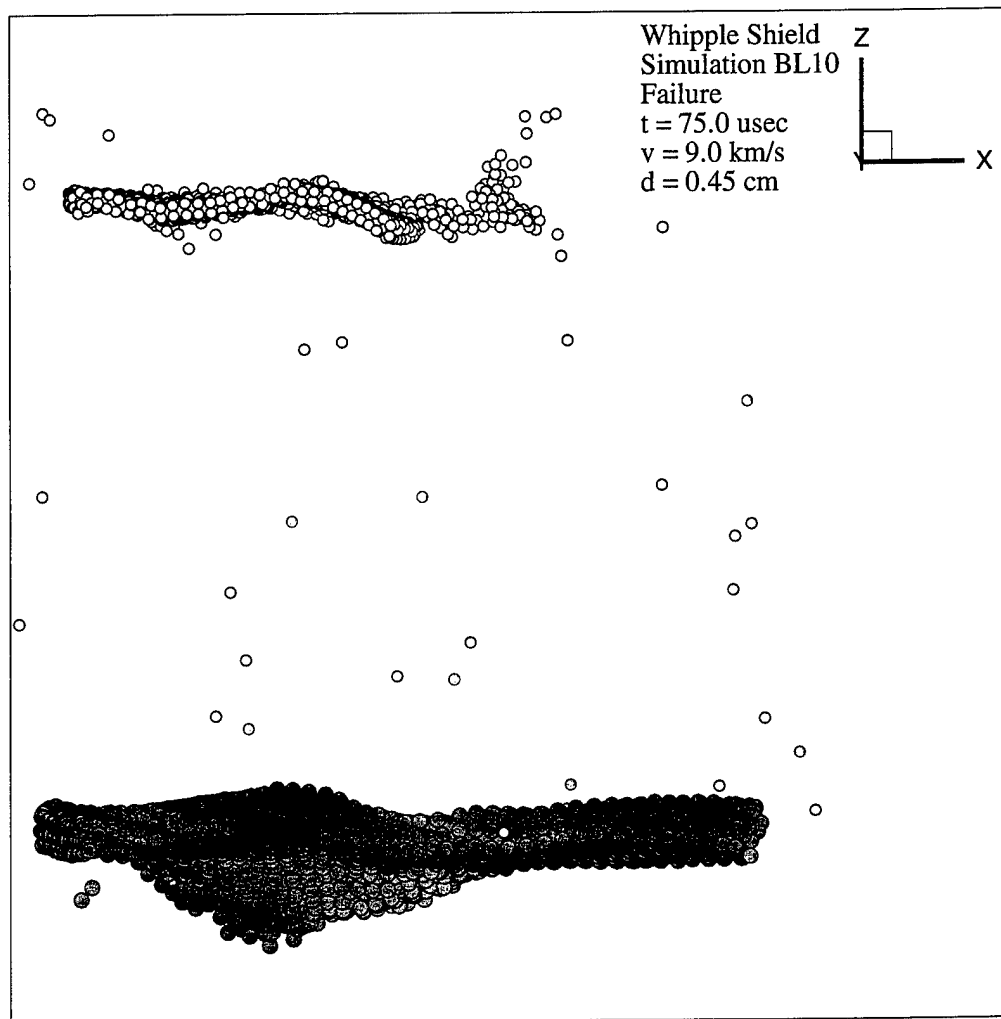


Figure 7: Side View of 15° EXOS Result for Simulation BL10

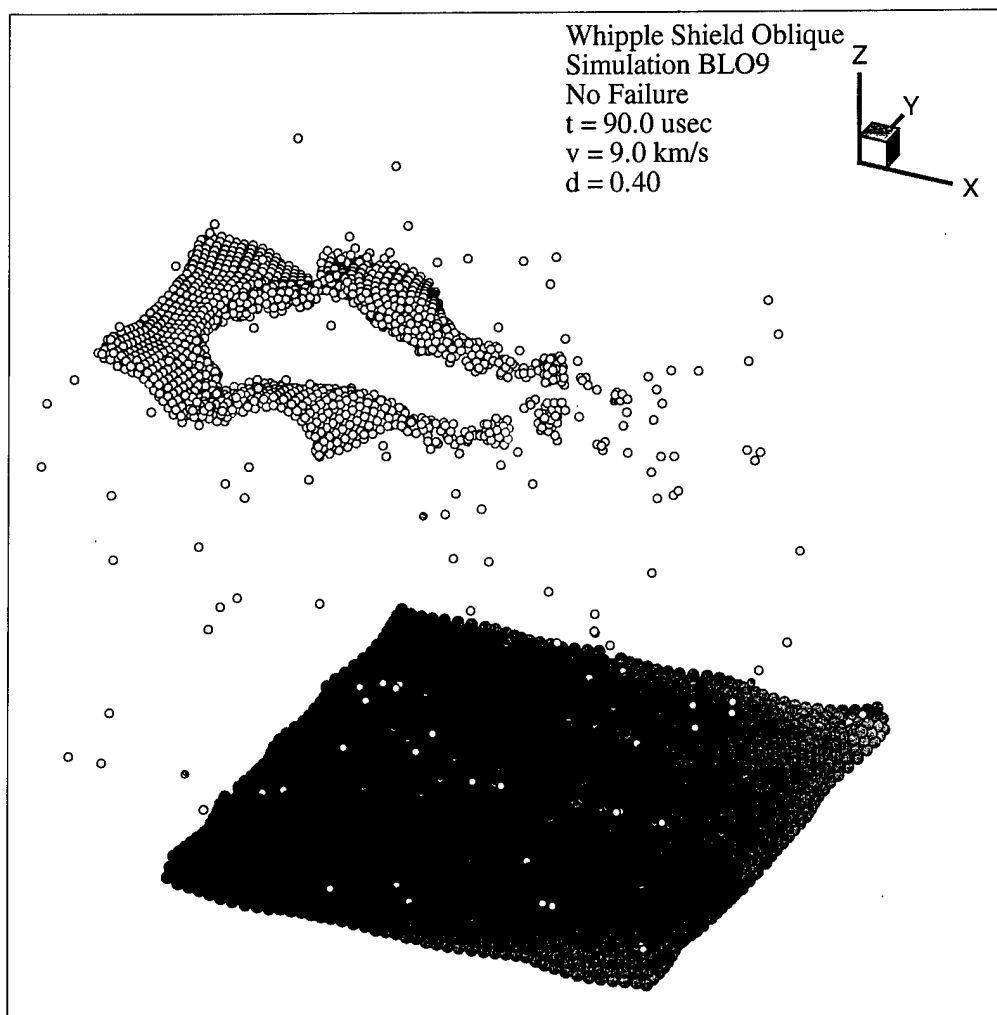


Figure 8: 45° EXOS Result for Simulation BLO9

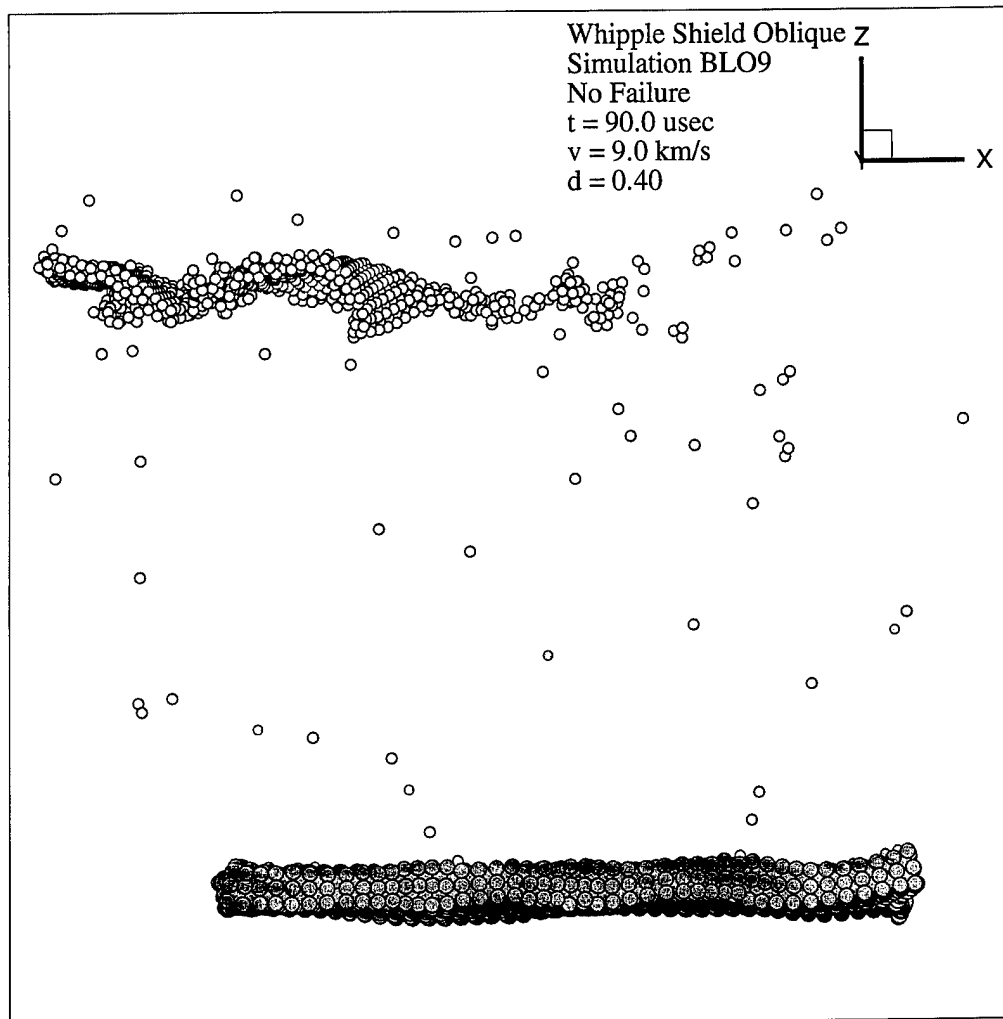


Figure 9: Side View of 45° EXOS Result for Simulation BLO9

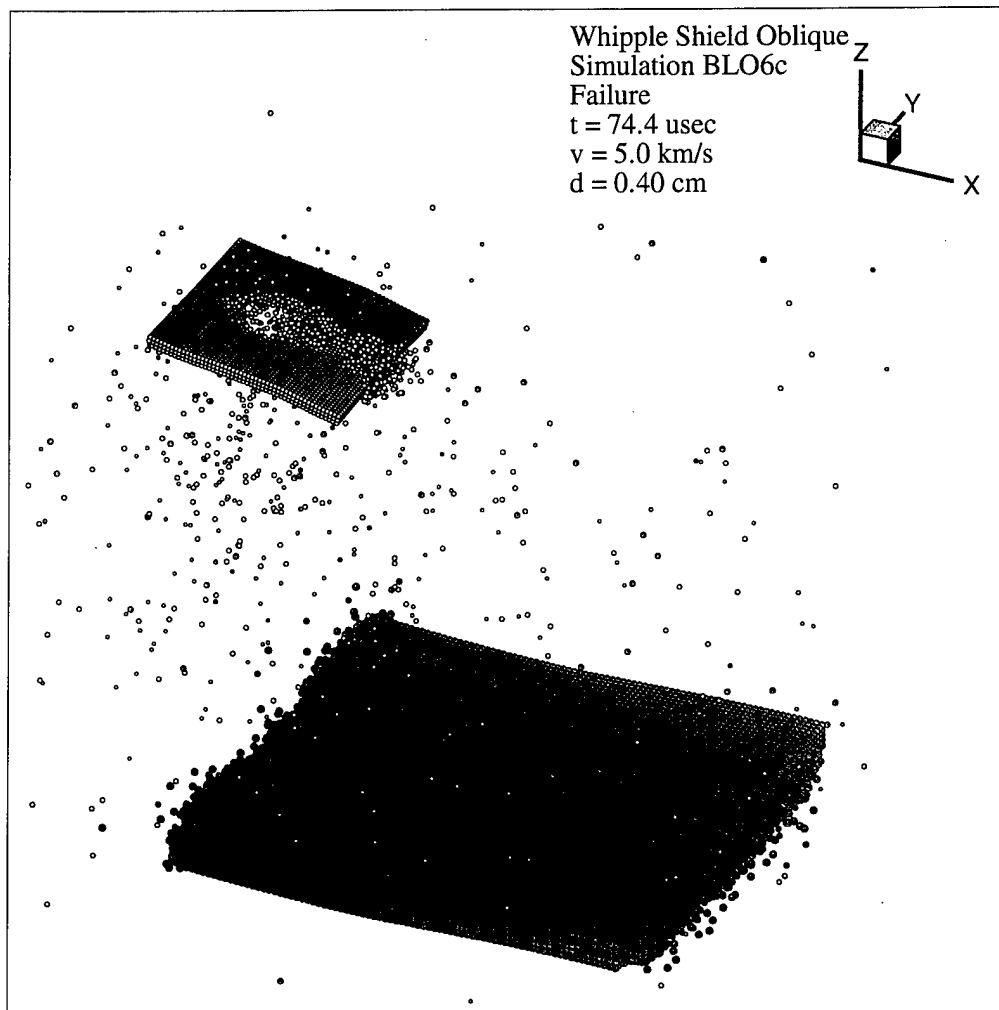


Figure 10: 45° EXOS Result for Simulation BLO6c

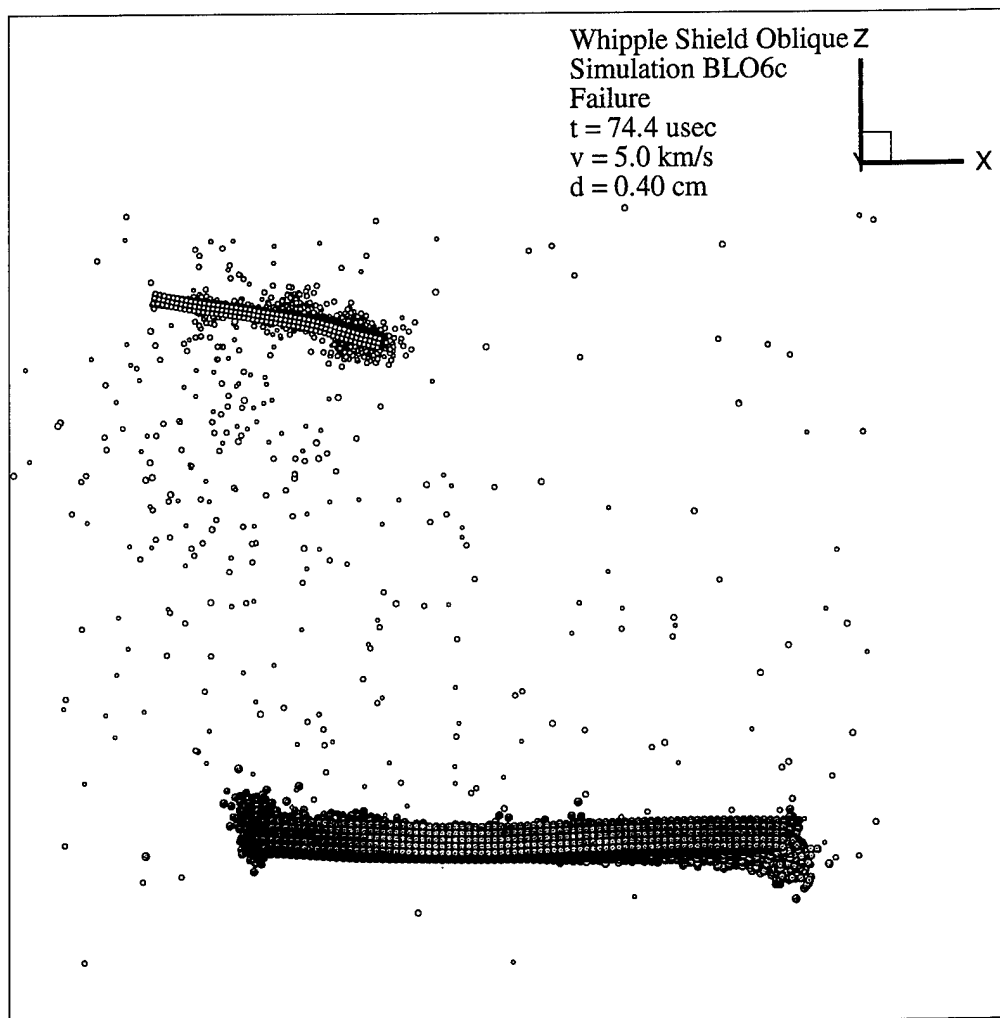


Figure 11: Side View of 45° EXOS Result for Simulation BLO6c



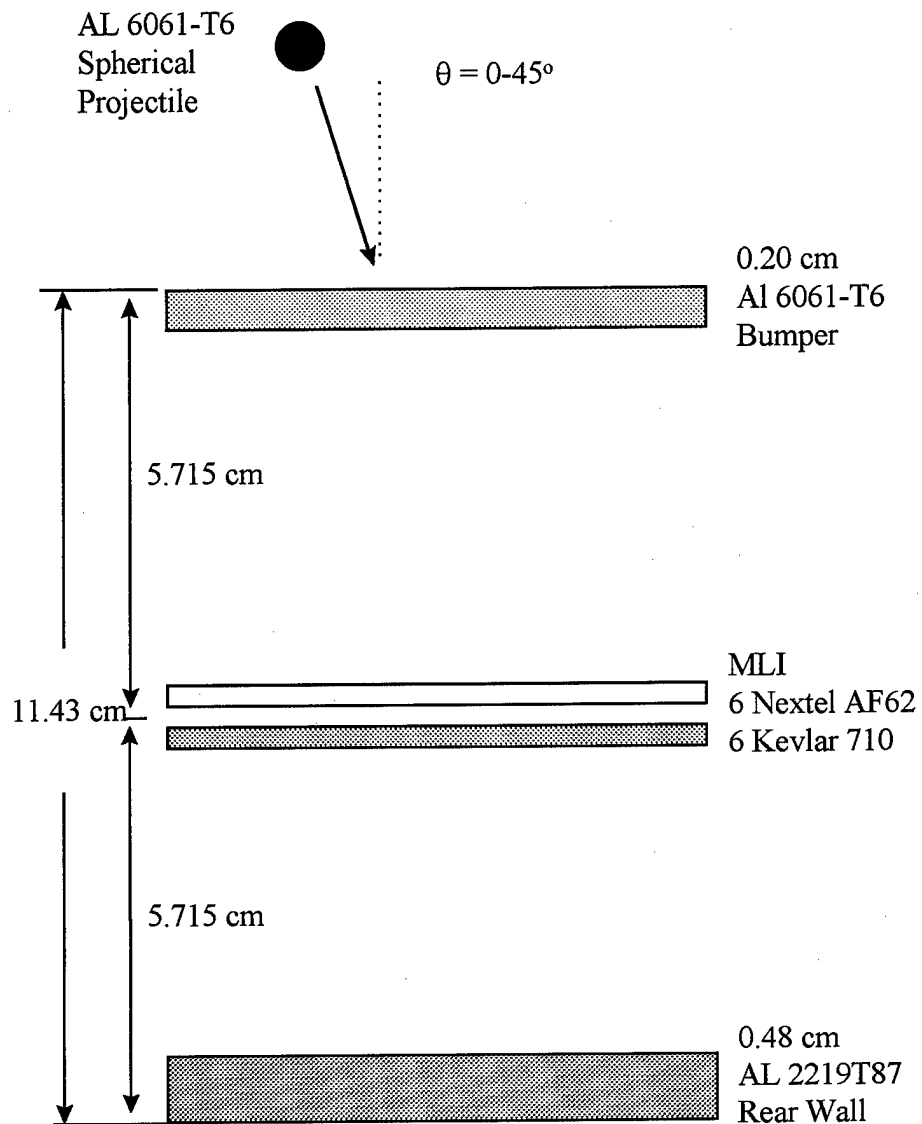


Figure 12: Stuffed Whipple Shield Geometry

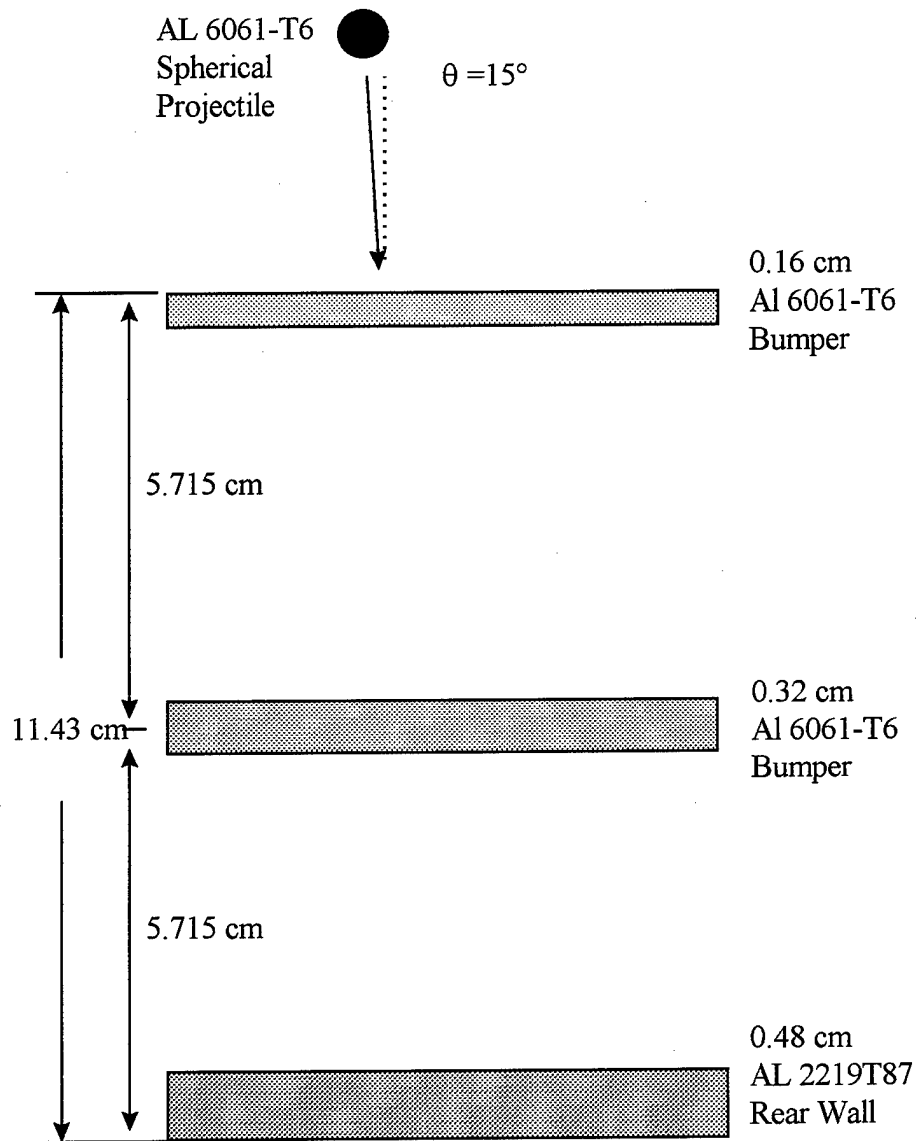


Figure 13: All Aluminum Stuffed Whipple Shield Geometry

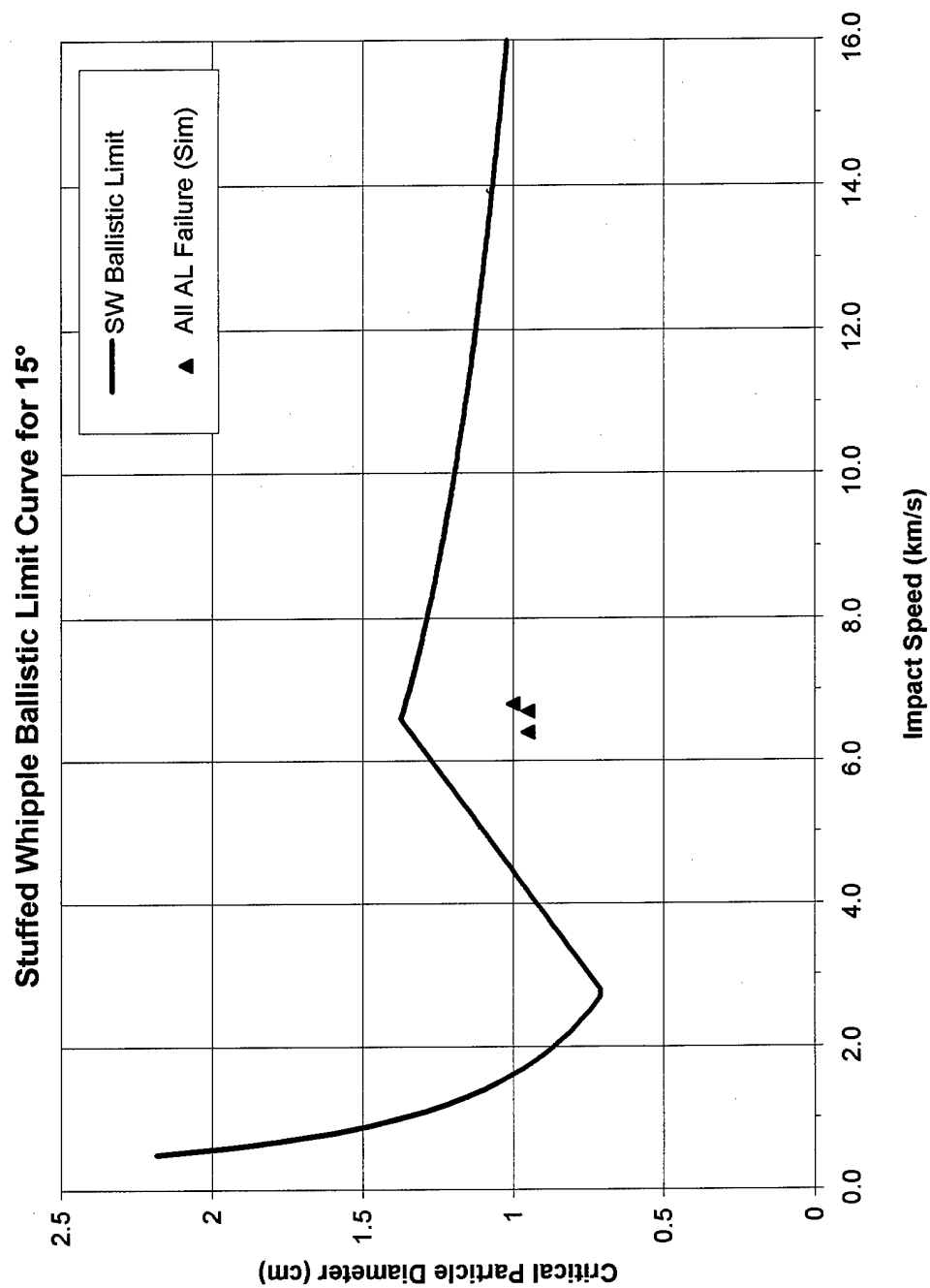


Figure 14: Ballistic Limit Curve for the 15° Stuffed Whipple Shield Simulation

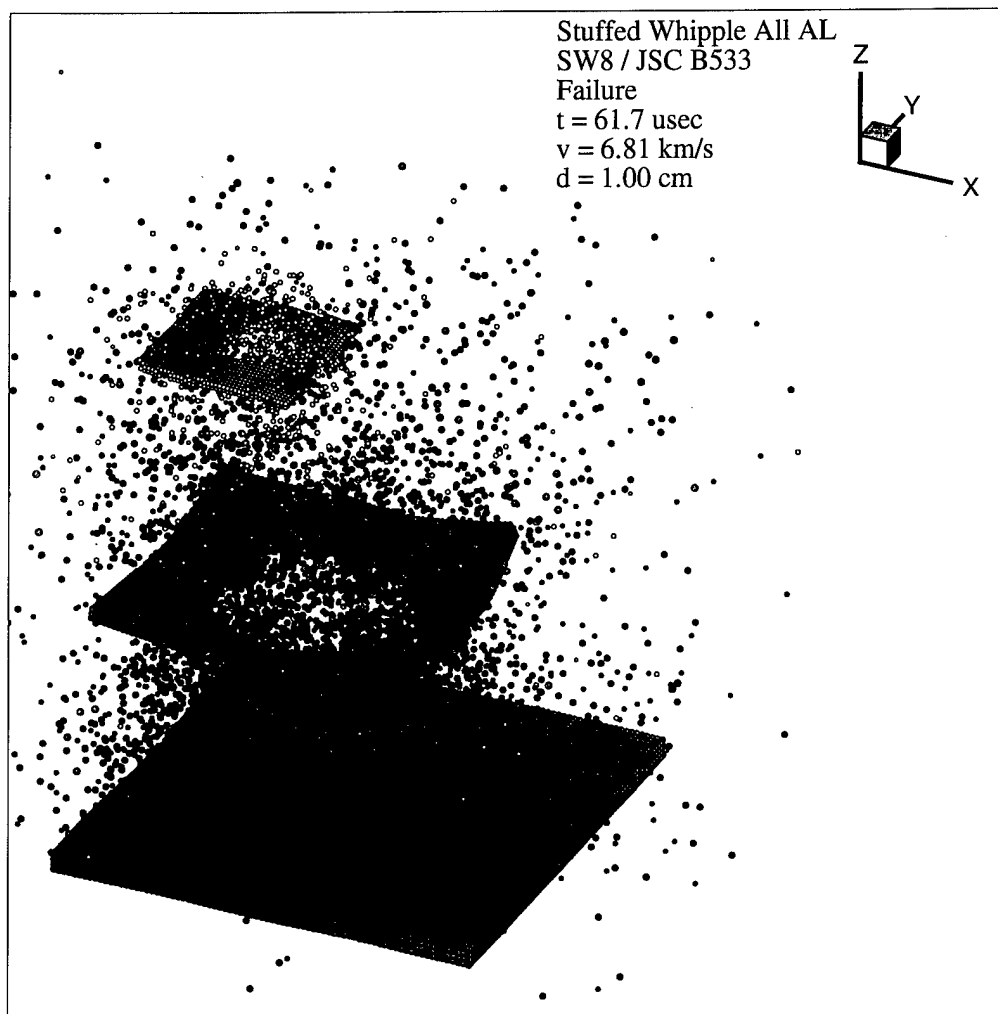


Figure 15: Stuffed Whipple Shield EXOS Result for Simulation SW8

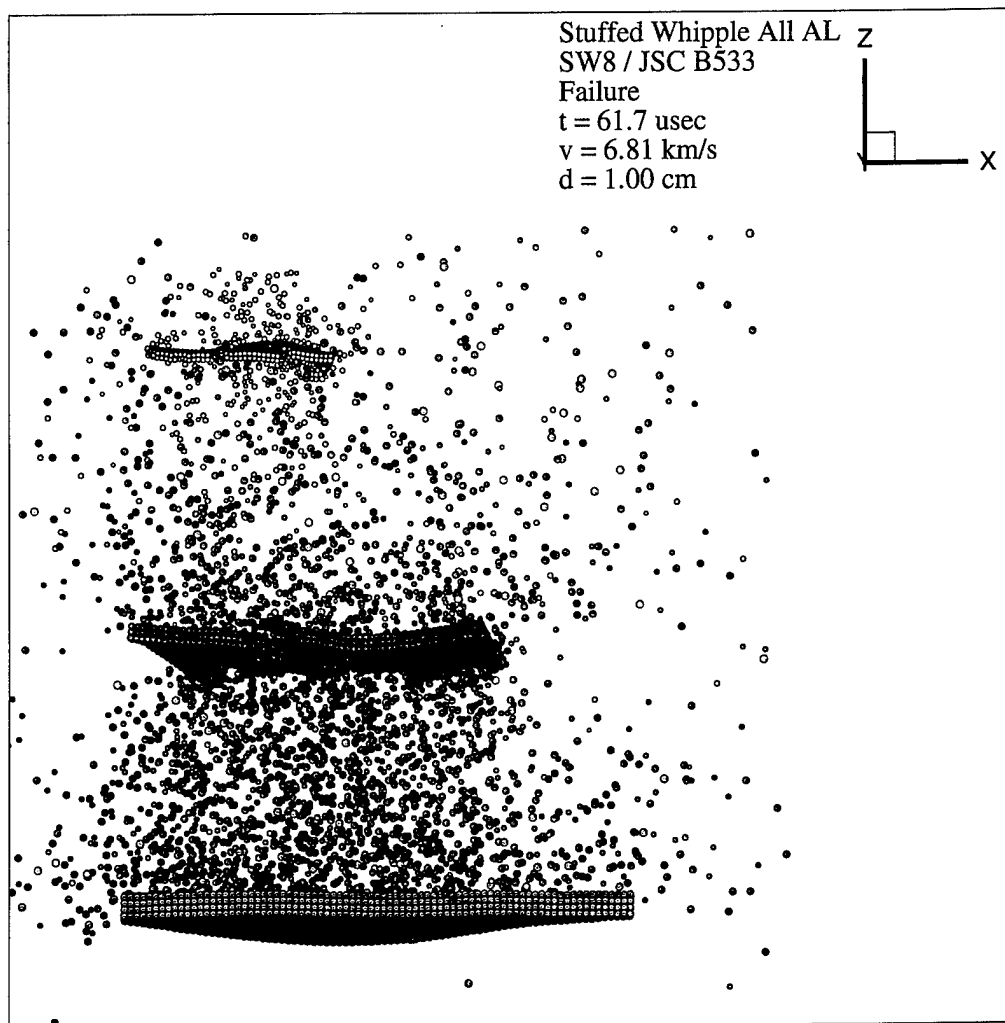


Figure 16: Side View of Stuffed Whipple Shield EXOS Result  
for Simulation SW8

## Appendix A: EXOS Input File

BLO8c- 7.0/0.450

control data

	0.0	0	1	0	1
8000	12000	100.0	0.000	10000.0	
1.0e-9	10.0	0.0	1.0		
3	-4.50				

material data

3					
1	4				
0.29e-2	1.0e-00	1.00	0.885e-2	2.703	.293e-0
1.00e-1	0.276e-0	0.58e-2	1.0e-02	3.00e+0	0.012e+1
0.10e-1	0.885e-2	0.1e-1	0.5240	1.400	1.97
2	4				
0.29e-2	1.0e-00	1.00	0.885e-2	2.703	.293e-0
1.00e-1	0.276e-0	0.58e-2	1.0e-02	3.00e+0	0.012e+1
0.10e-1	0.885e-2	0.1e-1	0.5240	1.400	1.97
3	4				
0.29e-2	1.0e-00	1.00	0.885e-2	2.703	.293e-0
1.00e-1	0.276e-0	0.58e-2	1.0e-02	3.00e+0	0.012e+1
0.10e-1	0.885e-2	0.1e-1	0.5240	1.400	1.97

sphere

1			
0.000	0.4500	0.0	
0.4950	-0.4950	0.0	
0.2250	2.703	0.885e-2	0.0
3			
0			

plate

2			
0.750	-0.0635	0.000	
0.0	0.0	0.0	
0.06350	2.703	0.885e-2	0.0
50	2	50	
1.0	1.0	1.0	
0			

plate

3			
3.600	-5.3658	0.000	
0.0	0.0	0.0	
0.0794	2.703	0.885e-2	0.0
80	4	80	
1.0	1.0	1.0	
0			

## Appendix B: Rezoner Input File

itype	0	
mttype	0	
xmin	1	-1.75
xmax	1	7.00
ymin	1	-6.75
ymax	1	0.75
zmin	1	-3.75

## Bibliography

- Anderson, C.E., Jr., 1987, "An Overview of the Theory of Hydrocodes," *International Journal of Impact Engineering*, Vol. 5, pp. 33-59.
- Belytschko, T., and Lin, J.I., 1987, "A Three-Dimensional Impact-Penetration Algorithm with Erosion," *International Journal of Impact Engineering*, Vol. 5, pp. 111-127.
- Benz, W., and Asphaug, E., 1995, "Simulations of Brittle Solids Using Smooth Particle Hydrodynamics," *Computer Physics Communications*, Vol. 87, pp. 253-265.
- Budge, K.G., and Peery, J.S., 1993, "RHALE: A MMALE Shock Physics Code Written in C++," *International Journal of Impact Engineering*, Vol. 14, pp. 107-120.
- Christiansen, E.L., 1993, "Design and Performance Equations for Advanced Meteoroid and Debris Shields," *International Journal of Impact Engineering*, Vol. 14, pp. 145-156.
- Christiansen, E.L., Crews, J.L., Williamsen, J.E., Robinson, J.H., and Nolen, A.M., 1993, "Enhanced Meteoroid and Orbital Debris Shielding," *International Journal of Impact Engineering*, Vol. 17, pp. 217-228.
- Christiansen, E.L., and Kerr, J.H., 1997, "Projectile Shape Effects on Shielding Performance at 7 Km/s and 11 Km/s," *International Journal of Impact Engineering*, Vol. 20, pp. 165-172.
- Davis, J.R., 1993, Aluminum and Aluminum Alloys, ASM International, Materials Park, Ohio.
- Fahrenthold, E.P., 1993, "A Lagrangian Model for Debris Cloud Dynamics Simulation," *International Journal of Impact Engineering*, Vol. 14, pp. 229-240.



- Fahrenthold, E.P., 1995, "Oblique Hypervelocity Impact Simulation for Whipple Shield-Protected Structures," *International Journal of Impact Engineering*, Vol. 17, pp. 291-302.
- Fahrenthold, E.P., 1998, User's Guide for EXOS, Version 1.25, University of Texas, Austin.
- Fahrenthold, E.P., and Koo, J.C., 1997, "Energy Based Particle Hydrodynamics for Hypervelocity Impact Simulation," *International Journal of Impact Engineering*, Vol. 20, pp. 253-264.
- Fair, H., 1987, "Hypervelocity Then and Now," *International Journal of Impact Engineering*, Vol. 5, pp. 1-11.
- Hertel, E.S., 1993, "Comparison of Analytic Whipple Bumper Shield Ballistic Limits with CTH Simulations," Sandia Report SAND92-0347.
- Isbell, W.M., and Tedeschi, W.J., 1993, "Hypervelocity Research and the Growing Problem of Space Debris," *International Journal of Impact Engineering*, Vol. 14, pp. 359-372.
- Johnson, G.R., Petersen, E.H., and Stryk, R.A., 1993, "Incorporation of an SPH Option into the EPIC Code for a Wide Range of High Velocity Impact Computations," *International Journal of Impact Engineering*, Vol. 14, pp. 385-394.
- Johnson, G.R., and Beissel, S.R., 1996, "Normalized Smoothing Functions for SPH Impact Computations," *International Journal for Numerical Methods in Engineering*, Vol. 39, pp. 2725-2741.
- Libersky, L.D., Petschenk, A.G., Carney, T.C., Hipp, J.R., and Allahdadi, F.A., 1993, "High Strain Lagrangian Hydrodynamics," *Journal of Computational Physics*, Vol. 109, pp. 67-75.
- Lu, Y.Y., Belytschko, T., and Tabbara, M., 1995, "Element-free Galerkin Method for Wave Propagation and Dynamic Fracture," *Computer Methods in Applied Mechanics and Engineering*, Vol. 126, pp. 131-153.
- McGlaun, J.M., Thompson, S.L., and Elrick, M.G., 1990, "CTH: A Three Dimensional Shock Wave Physics Code," *International Journal of Impact Engineering*, Vol. 10, pp. 351-360.

- Monaghan, J.J., 1985, "Particle Methods for Hydrodynamics," *Computer Physics Reports.*, Vol. 3, pp. 71-124.
- Monaghan, J.J., 1988, "An Introduction to SPH," *Computer Physics Communications.*, Vol. 48, pp. 89-96.
- Monaghan, J.J., 1992, "Smoothed Particle Hydrodynamics," *Annu. Rev. Astron. Astrophys.*, Vol. 30, pp. 543-574.
- Stellingwerf, R.F., and Wingate, C.A., 1993, "Impact Modeling with Smooth Particle Hydrodynamics," *International Journal of Impact Engineering*, Vol. 14, pp. 707-718.
- Sulsky, D., Chen, Z., and Schreyer, H.L., 1994, "A Particle Method for History Dependent Materials," *Computer Methods in Applied Mechanics and Engineering*, Vol. 118, pp. 179-196.
- Wingate, C.A., Stellingwerf, R.F., Davidson, R.F., and Burkett, M.W., 1993, "Models of High Velocity Impact Phenomena," *International Journal of Impact Engineering*, Vol. 14, pp. 819-830.

## **Vita**

Robert James Rabb is a native of Amarillo, Texas and was born on September 8, 1966, the son of Tram V.N. Rabb and Donald Fitzhugh Rabb. After completing his work at Caprock High School, Amarillo, Texas, in 1984, he entered the United States Military Academy in West Point, New York. He received the degree of Bachelor of Science in May, 1988 and was commissioned as an officer in the United States Army. During the following years, he served in Operation Desert Storm in Saudi Arabia and in Operation Joint Endeavor in Bosnia and Croatia. In June, 1996, he entered The Graduate School at the University of Texas, enroute as an instructor to the United States Military Academy.

Permanent address: 3201 S. Spring  
Amarillo, Texas 79103

This thesis was typed by the author.